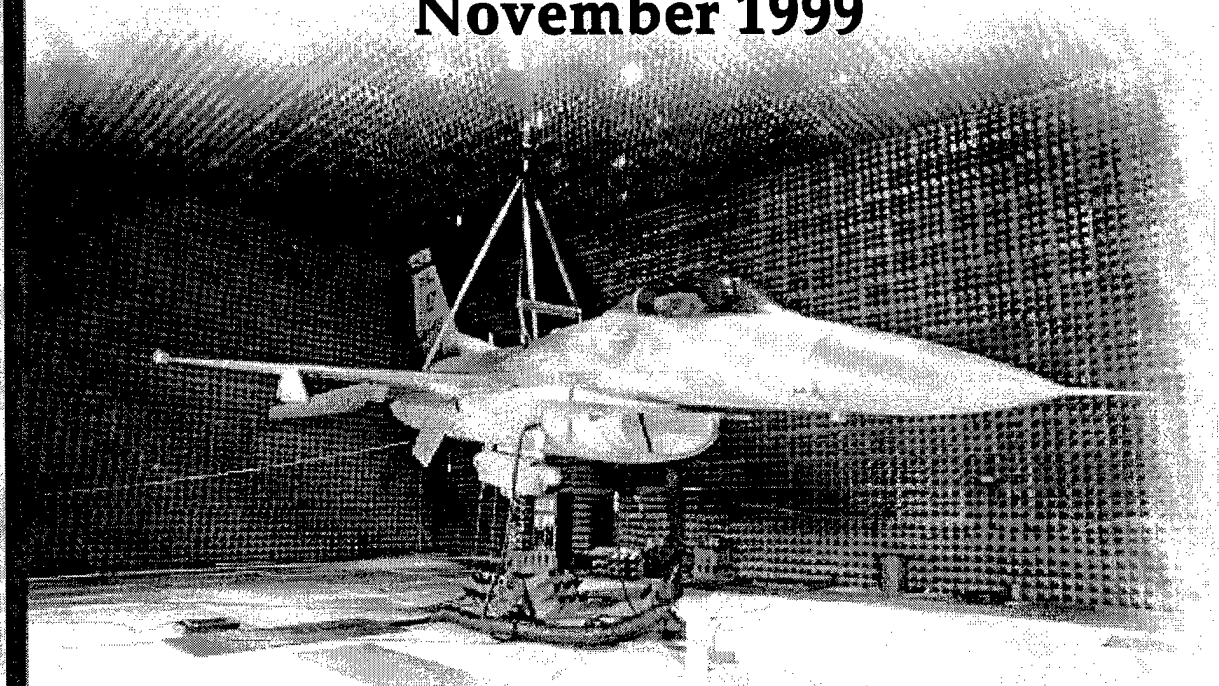
JADS JT&E-TR-99-015

JADS JT&E

Electronic Warfare Test Interim Report Phase 3

By: Maj Darrell L. Wright and
Capt Roman M. J. Nation

November 1999



Joint Advanced Distributed Simulation Joint Test Force . 2050A 2nd St. SE . Kirtland AFB, NM 87117-5522

Table of Contents

Appendices

List of Figures

List of Tables

Executive Summary

1.0 Introduction

This summary serves as a stand-alone document, as well as part of this report. Therefore, there is some duplication between this summary and the full report.

2.0 JADS Overview

The Joint Advanced Distributed Simulation (JADS) Joint Test and Evaluation (JT&E) was chartered by the Deputy Director, Test, Systems Engineering and Evaluation (Test and Evaluation)[1], Office of the Secretary of Defense (Acquisition and Technology) in October 1994 to investigate the utility of advanced distributed simulation (ADS) technologies for support of developmental test and evaluation (DT&E) and operational test and evaluation (OT&E). The JADS Joint Test Force (JTF) is Air Force led with Army and Navy participation. The JADS JT&E program is scheduled to end in March 2000.

The JADS JTF investigated ADS applications in three slices of the test and evaluation (T&E) spectrum: ADS support of air-to-air missile testing; ADS support for command, control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR) testing; and the Electronic Warfare (EW) Test which explored ADS support for EW testing.

3.0 EW Test Overview

The tasking to conduct an ADS-based EW test called for an airborne self-protection jammer (SPJ) as the system under test (SUT). The emphasis of the EW Test was on the performance of the ADS components and their contribution or impact to testing rather than on the performance of the SPJ pod itself. Measures of performance (MOPs) for the SPJ were identified as measures that would most likely be affected by distributed testing. Statistical comparison of the MOPs became the methodology for evaluating ADS. JADS evaluated distributed test control and analysis, network performance, relationships between data latencies, and ADS-induced data anomalies. Time, cost, and complexity, as well as validity and credibility of the data, were part of the evaluation.

The EW Test was designed as a three-phase effort. The first phase provided a baseline of jammer performance data in a non-ADS environment that was then compared to the data collected in the second and third phases using an ADS environment. The second phase used a digital system model of the SPJ representing an early developmental test. The third phase used the SPJ mounted on the aircraft that was suspended in an installed systems test facility (ISTF). This test

[1] This office is now the Deputy Director, Developmental Test and Evaluation (DD, DT&E).

represented a combined integration and effectiveness test that would occur late in the SPJ development.

Phase 1 included a risk reduction flight test effort at the Western Test Range (WTR) to define a reference test condition; 14.4 hours of baseline flight test using a modified ALQ-131 jamming pod at the WTR; a nine-day hardware-in-the-loop (HITL) test at the Air Force Electronic Warfare Environment Simulator (AFEWES) at Fort Worth, Texas; and a three-day system integration laboratory (SIL) test at the Automatic Multiple Environment Simulator (AMES) facility at Eglin Air Force Base (AFB), Florida. The HITL and SIL tests were added to supplement the baseline flight testing and to provide missing data. This established the baseline of environment and jammer performance data against two command-guided surface-to-air missile (SAM) sites, one semiactive surface-to-air missile site, and one anti-aircraft artillery (AAA) site. The reference test condition and baseline data were used to develop the ADS test environment for the two subsequent ADS test phases and provided the baseline data for comparison with the ADS test results.

Phase 2 was a test of a real-time digital system model (DSM) of the modified ALQ-131 receiver processor linked with terminal threats at the AFEWES facility and a scripted model of the terminal threat hand-off portion of an integrated air defense system (IADS). The reference test condition used in the Phase 1 flights was replicated as closely as possible in the synthetic ADS environment; the jammer model was flown, via the scripted flight profiles developed from the actual open air range (OAR) baseline flights and HITL test, against the AFEWES threats positioned in the synthetic environment as the threats were positioned on the range.

Phase 3, the subject of this report, used the modified ALQ-131 jammer installed on an F-16 aircraft in the Air Combat Environment Test and Evaluation Facility (ACETEF) located at Patuxent River Naval Air Station, Maryland. This facility was linked with AFEWES threats using the same reference test condition as previous tests and controlled by the same scripted flight profile and rules of engagement.

4.0 Overview of EW Test Phase 3

4.1 Purpose

The primary purpose of Phase 3 was to collect SPJ performance data using an ALQ-131 in an ADS-based test environment. The performance data were combined with data collected on the ADS environment itself to determine how much of an impact ADS had on the test. Phase 3 test objectives are summarized in Table ES-1.

Table ES-1. Test Objectives

Obj #	Objective
1-1	Establish SPJ performance in JADS/ISTF environment
1-2	Establish the repeatability of ISTF test results
1-3	Establish ranges of ISTF statistics for event data
1-4	Establish range of correlation coefficients for series observables
1-5	Quantify the effects of data latency on JADS/ISTF test environment
1-6	Quantify the operating reliability and mean time between failure of the JADS network
1-7	Determine the connectivity performance of the JADS network

4.2 Approach

The overall test approach was designed to provide a means of capturing the ADS effects within the Phase 3 architecture. The high level architecture (HLA) was used to link the SPJ located at ACETEF, HITL terminal threats at the AFEWES facility, and other models hosted in the JADS test control facility. The test collected data for subsequent comparison with the EW Test MOPs collected in Phase 1 and Phase 2 as well as the ADS data needed to calculate ADS MOPs. A statistical comparison was used to compare the EW Test MOP data sets. The results of the specific EW Test MOPs are classified and are reported in a separate document. The statistical comparisons of the MOPs resulted in a correlation measure called a "P-value." P-values are unclassified and are included in this report, but should be reviewed in the context of the classified report for better understanding.

5.0 Phase 3 Test Results

5.1 General Results

Phase 3 used an HLA-compliant ADS architecture to successfully recreate both an open air test and hardware-in-the-loop test. The architecture successfully integrated an ALQ-131 self-protection jammer installed in an anechoic chamber with the high fidelity threats at AFEWES. This implies that ADS may be used to address the EW test process limitations. A complete discussion on the utility of ADS to EW testing will be the subject of the JADS EW Test final report.

Test results and operator interviews indicated that even though there were some isolated incidents of ADS impacting results, there were no consistent ADS-induced biases or flaws that made the data invalid. Data latency in excess of the design goal and lengthy bursts of lost aircraft position data did not affect the EW Test MOPs in any consistent, measurable fashion. Subject matter experts confirmed that the data produced by the JADS architecture were valid. This implies that properly designed ADS architectures will not impact test results. Because of improvements made in the runtime infrastructure (RTI) after the Phase 2 test, the ADS MOPs showed improved performance (e.g., less average latency, less overall data loss, etc.) over the Phase 2 results.

There were limitations within the ADS architecture that JADS created. Different jammer techniques and more reactive players required that the bursts of lost aircraft position data be resolved and latency performance be improved over what was observed in Phase 2. Predictive jammer techniques would also require more of the jammer processing logic to be collocated at AFEWES. Several of the message structures and common data used in our test would have to be examined before being used in other tests. While all the message structures have room for growth, they need to be examined by future implementers to ensure the size and intent meet the requirements of the new federation.

The most significant limitation to this architecture was the availability of threats suitable for ADS-based testing. Low fidelity threats are not difficult to add to this architecture, but they must run in real time to interact properly with the manned threats and the SPJ. Models are not sufficient to address shortfalls of the EW test process since they do not recreate the largest source of variation - human operator actions. Manned high fidelity threat representations are the key to obtaining the highest benefit from this architecture. The AFEWES facility uses distributed simulation techniques within its facility to accomplish traditional testing. ADS simply allows AFEWES to connect to other facilities or locations. The OAR used in Phase 1 had high fidelity threat simulators as well. However, these were not suitable in their current configuration to accomplish testing within the JADS architecture. Radio frequency injection into the threat for both target and jamming is key to making these threat assets available using ADS.

6.0 Conclusions

Phase 3 demonstrated that ADS tests create valid EW test data when properly designed. ADS can be used to connect ISTFs with manned threat simulators. This makes ADS a potentially feasible tool for EW testers. However, the availability of suitable manned simulators will likely determine how quickly ADS is integrated into the mainstream of EW testing.

1.0 Introduction

1.1 Overview

The Joint Advanced Distributed Simulation (JADS) Joint Test and Evaluation (JT&E) program is an Office of the Secretary of Defense (OSD)-sponsored joint-service effort designed to determine how well advanced distributed simulation (ADS) can support test and evaluation (T&E) activities. The Electronic Warfare (EW) Test was one of three tests comprising the JADS Joint Test Force (JTF). It was chartered separately in 1996 to test the utility of distributed simulations to the EW T&E community. This report focuses on results of the EW Test Phase 3 system testing using the high level architecture (HLA).

The JADS EW Test was designed to provide insight into ADS-based testing for JADS. JADS was charted to address three issues.

- What is the present utility of ADS for T&E?
- What are the critical constraints, concerns, and methodologies when using ADS for T&E?
- What are the requirements that must be introduced into ADS systems if they are to support a more complete T&E capability in the future?

These issues were mapped to activities within the EW Test effort. This mapping first appeared in the 1996 JADS Analysis Plan for Assessment (APA). Further refinement was described in the 1997 JADS Program Level Test Analysis Plan (TAP)/Data Management and Collection Plan (DMAP). Execution of the Phase 3 test brought further refinements that are summarized later in this report.

Additionally, ADS was expected to bring specific benefits to the EW test process. The EW test process is a formally documented, systematic test process covering all phases of system development. It served as the template for the System Test and Evaluation Process (STEP) adopted by OSD. During the JADS Feasibility Study, three shortfalls in implementing the EW test process were identified that ADS might solve. These shortfalls are discussed below.

The first shortfall is the inability to correlate test results throughout the development process. The EW test process recommends a model, test, model approach. Limitations in both facilities and models result in fidelity differences in both the system under test (SUT) and the threats because of their continuing evolution. Too many variables change between test events to trace apparent performance changes that may be due to threat differences between facilities or SUT design evolution. ADS holds the promise of allowing a fixed set of high fidelity threats to be used throughout the development process. Limiting the threat representation to one set of high fidelity threats implies all performance differences would be due to SUT evolution. This, in turn, would allow statistical comparisons to provide decision makers with a better understanding of system performance.

The second shortfall, test resource fidelity, also relates to correlation. High fidelity test resources such as man-in-the-loop threat simulators are expensive and therefore available at very few

facilities and duplication of the highest fidelity resources is minimal. The tester is often forced to use low fidelity threat simulators or models early in system development. Previous testing against high fidelity threats required the system to be transported to the appropriate facility and integrated. Transportation is often impractical with breadboard and brassboard hardware. Testing against models prohibits determination of how the SUT affects operator actions. For self-protection jammers (SPJs), this interaction is critical. ADS holds the promise of allowing the SUT of variable fidelity to interact with the high fidelity threat resources without collocating them. This would allow early representations of lower fidelity SUT resources to interact with high fidelity threat resources. Any real-time representation of the SUT, including digital system models, could be used for testing. This would allow system designers to see critical interactions, such as operator actions, very early in the design process.

The third shortfall is resource availability. Test facilities are limited by budget realities that force them to limit testing to specific capabilities at limited times. There is no single test facility that provides the tester with all the high fidelity resources and other support needed to completely test complicated EW systems. This is especially true of jammer systems. ADS holds the promise of allowing the tester to link together the resources needed to accomplish the test no matter where the resource is located. This would allow traditionally separate tests to be conducted in coordination with one another.

The EW Test was designed around three test phases to address both the JADS issues and the ability of ADS to solve the three EW test process shortfalls discussed above. Phase 1 used traditional test methods and environments to establish a performance baseline of an operational airborne SPJ against four threats. This phase was accomplished in three different environments. These separate environments were needed to overcome test instrumentation limitations and procedure problems that prevented JADS from measuring all the performance measures in a single environment. Jammer effectiveness measures were collected in both the open air range (OAR) and Air Force Electronic Warfare Evaluation Simulator (AFEWES) facilities in Fort Worth, Texas. Jammer internal response times were measured in a system integration lab (SIL). These results are reported in both classified and unclassified formats (see http://www.jads.abq.com). These reports focus primarily on non-ADS test execution.

Phase 2 used a digital system model (DSM) to represent the jammer. The DSM was hosted at Air Combat Environment Test and Evaluation Facility (ACETEF) Patuxent River, Maryland, geographically separated from the threats at AFEWES, but it interacted with the threats to recreate the baseline data. Great care was taken to ensure the same reference test condition was used between the different test phases. Several of the key components of the Phase 2 test were scripts built from actual OAR recorded data or developed from the reference test condition flown on the range. Statistical correlation of the EW Test measures of performance (MOP) would be used to compare the ADS results with the traditional test results obtained in Phase 1. The correlation was expected to provide insight into how much ADS impacted the test results.

Phase 3, used the same components as Phase 2 except for the DSM. A real jammer installed in the ACETEF facility replaced the DSM in this test phase. The real jammer required JADS to use the ACETEF gateway to allow radio frequency (RF) stimulators to recreate the signals of the RF

environment for the jammer. The same EW Test MOP were collected as in the previous phases. Statistical correlation of the EW Test MOP was used to compare the ADS results with one another and with the traditional test results. The correlation provided insight into how well the ADS results could be repeated and how much ADS impacted the test results. The Phase 3 test results are the subject of this report.

Phase 2 did not provide complete answers to the issues addressed by the EW Test. An interim report was issued that discussed test execution, unclassified ADS measure results, unclassified correlation results, and lessons learned. The complete answer to the JADS issues and the ability of ADS to address the EW test shortfalls is presented in this report. A complete presentation of the EW Test MOP results is contained in a separate classified report. This report will also address the EW test shortfalls to provide the EW community with a single reference source.

Additional background information on the history and planning for the EW Test in general and the Phase 3 effort specifically is contained in the references listed below.

1.2 References

Electronic Warfare Test Analysis Plan for Assessment, May 1996

Program Level TAP/DMAP, March 1998

Electronic Warfare Phase 3 TAP/DMAP, February 1999

Electronic Warfare Test Interim Report Phase 1, March 1999

Electronic Warfare Test Phase 1 Classified Results Report, September 1999

Electronic Warfare Test Interim Report Phase 2 , September 1999

Electronic Warfare Test Classified Results Report, projected for November 1999

1.3 Electronic Warfare Test

The tasking to conduct an ADS-based test of an EW system specifically called for the use of an airborne SPJ as the surrogate system under test (SUT). The actual SUT for JADS was ADS. In the summer of 1995, JADS presented a comprehensive test and analysis approach for an EW Test to the JT&E Technical Advisory Board (TAB) and the Senior Advisory Council (SAC). The JADS EW Test approach was fully supported by the TAB but not chartered initially primarily because of the high cost, $18 million. In response, JADS tailored the initial EW Test design and subsequently developed a reduced scope, lower cost EW test and analysis approach using a modified ALQ-131 SPJ pod as the surrogate SUT. For this test, the ALQ-131 was modified to operate with tailored preflight and mission software tapes that affected its operational performance by limiting the signals responded to and modifying the signal from the pod from

optimum. Jamming effectiveness results during the JADS EW Test should not be taken as representative of operational pods in the tactical inventory.

The emphasis of the EW Test was on the performance of the ADS components and their contribution to testing rather than on the performance of the modified ALQ-131 test item itself. MOPs for the jammer were used as a means of evaluating ADS. These measures are listed in Table 1. JADS evaluated distributed test control and analysis, network performance, relationships among data latencies, and ADS-induced data anomalies. Time, cost, and complexity, as well as validity and credibility of the data are part of the evaluation. Specific reference test conditions were selected to allow this comparison. Additionally, some test activities that would not be feasible without ADS technology were planned.

Table 1. EW Test Measures of Performance

MOP #	Description
1	Correct threat identification
2	Correct threat identification response time
3	Correct electronic countermeasures (ECM) technique selection
4	Correct ECM technique selection response time
5	Jamming-to-signal ratio
6	Root mean square (RMS) tracking error
7	Number of breaklocks
8	Reduction in engagement time
9	Reduction in missiles launched
10	Missile miss distance

1.3.1 EW Test Approach

The EW Test was designed as a three-phase effort providing a baseline of SUT performance data in a non-ADS environment that was then compared to multiple tests of the same configuration in an ADS environment. The high level architecture (HLA) was used in the latter two phases.

Phase 1 included (1) an open air range (OAR) risk reduction flight test effort, (2) baseline flight test using a modified ALQ-131 jamming pod at the Western Test Range (WTR), (3) a hardware-in-the-loop (HITL) test at the Air Force Electronic Warfare Environment Simulator at Fort Worth, Texas, and (4) a system integration laboratory (SIL) test at the Air Warfare Center (AWC), Eglin Air Force Base, Florida. The purpose of this test phase was to establish a baseline of environment and SUT performance data against two command-guided surface-to-air missile (SAM) sites, one semi-active surface-to-air missile site, and one anti-aircraft artillery (AAA) site. This scenario was used to develop the ADS test environment for the following phases and provided the baseline data for comparison with the ADS-based test results. Additionally, the performance data provided a baseline for attempting to correlate the data across all three phases of the test. The test scenario was structured and constrained to provide the greatest opportunity for repeatability and therefore, good correlation.

The ADS-based phases, Phases 2 and 3, tested a real-time digital system model (DSM) representing the SUT (Phase 2) and the installed modified ALQ-131 on an F-16 (Phase 3) located in an integrated systems test facility (ISTF), respectively. The simulated threat environment and engagements closely resembled the OAR test. The baseline data collected in Phase 1 were used to create the synthetic replication of the aircraft as well as the engagement conditions.

1.3.2 EW Test Objectives

It is difficult to measure ADS utility in the real world of EW T&E. There are significant technical challenges in implementing ADS in the EW environment as well as programmatic issues such as cost and schedule impacts. The achievable (not just theoretical) performance that can be obtained by inserting ADS into the established EW test process must be determined. The overall objective of the JADS EW Test was to address these questions and thus assess the utility of ADS for EW test and evaluation. Specific test objectives are listed in the JADS EW Test APA, and Program Level TAP/DMAP.

2.0 Phase 3 Overview

2.1 Purpose

The purpose of the Phase 3 test was to collect SPJ performance data using the jammer in an ADS-based test environment. The HLA, specifically the runtime infrastructure (RTI) was used to link the jammer located at ACETEF, HITL terminal threats at the AFEWES facility, and other models and test control hosted in the JADS facility. The test collected data for comparison with MOP data collected in phases 1 and 2. Specifically, the analysis of Phase 3 results included descriptive statistics on the jammer and ADS MOPs; the repeatability analysis performed to check for repeatability within and across test phases; and the correlation process which compared each phase of test results to the others. Descriptions of each analysis process are published in the *EW Test Classified Results Report.* Repeatability and correlation results are explained and presented in Section 6 of this report as well as the *EW Test Classified Results Report.*

2.2 Organizational Structure

Figure 1 shows the organizational structure for coordinating and reporting during Phase 3 of the EW Test.

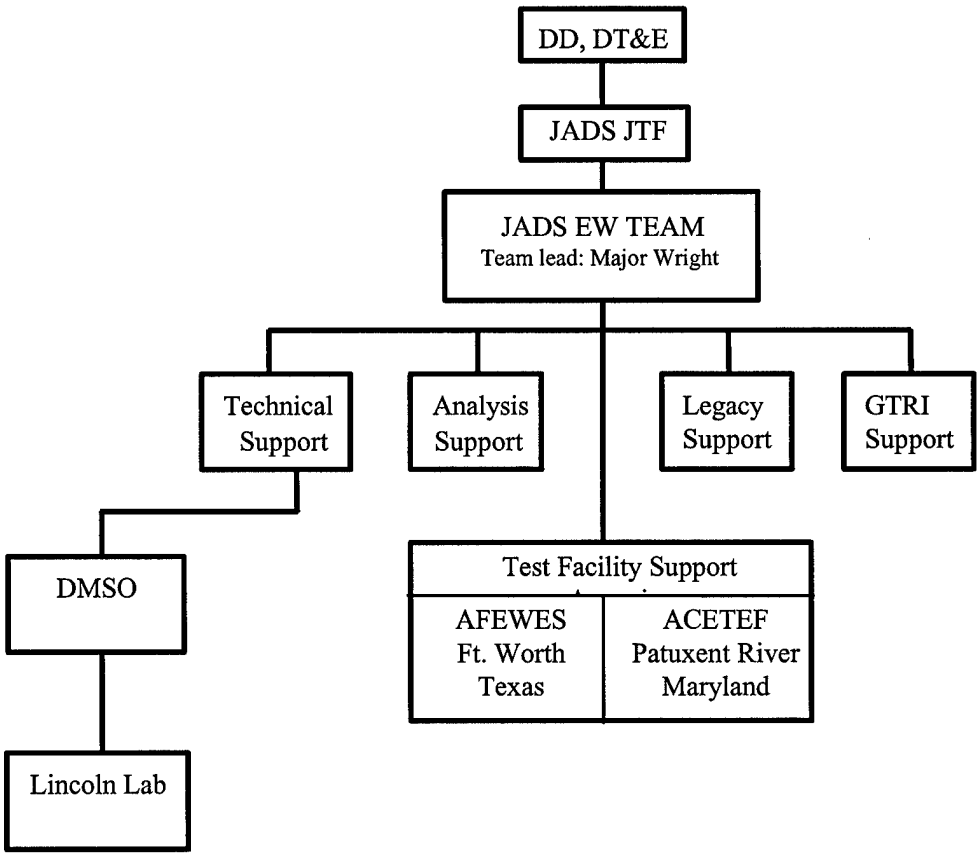


Figure 1. Organizational Structure

2.2.1 Roles and Responsibilities

The following sections detail the roles and responsibilities of each organization throughout the test design and execution of the ADS-based test phases.

2.2.1.1 Deputy Director, Developmental Test and Evaluation (DD, DT&E)

- Oversaw the JADS JT&E
- Approved the program test plan (PTP)
- Approved JADS financial requirements
- Oversaw the analysis and reporting of test results

2.2.1.2 JADS JTF and EW Test Team, Albuquerque, New Mexico

- Developed the Federation Execution Planners Workbook (FEPW)
- Developed the federation objective model (FOM)
- Developed the data logger software
- Managed the interface control document (ICD)
- Developed ADS measures and identified related data elements
- Acquired, installed, and supported communications routers, hubs, and switches
- Implemented and conducted benchmarks of the RTI, computer, and communications architecture to support JADS latency requirements
- Managed funding to accomplish the test
- Acquired, verified, and supported usage of T-1 long haul communications circuits
- Developed software tools for analyzing data and processing logger files
- Installed and integrated computer capabilities in the Test Control and Analysis Center (TCAC) and other sites
- Developed and integrated the components of the EW Test environment
- Developed and provided AFEWES, ACETEF, and Georgia Tech Research Institute (GTRI) with the Phase 3 TAP/DMAP
- Coordinated, rehearsed, and controlled execution of Phase 3 test activities
- Operated the TCAC during tests
- Analyzed and evaluated Phase 3 data and measures of performance
- Performed correlation testing on Phase 3 data in comparison to other test phases
- Reported interim and final results to the Office of the Secretary of Defense

2.2.1.3 Air Force Electronic Warfare Evaluation Simulator (AFEWES) 412th Test Wing, Fort Worth, Texas

- Developed the AFEWES federate software, integrated the new federate with the RTI, federate logger software and other JADS federation components
- Provided Phase 3 test facilities and AFEWES test management personnel
- Provided the use of the Tactical Air Mission Simulator (TAMS)

- Provided simulated terminal threats and system operators
- Provided the JammEr Techniques Simulator (JETS) for radio frequency signals to the threats
- Provided threat test management centers (TMC) for data collection
- Provided data, videotapes, and strip charts for each simulator
- Provided subject matter expert (SME) for verification and validation (V&V) of the distributed environment
- Provided inputs to the development of the reporting HLA documentation for JADS JTF

2.2.1.4 Air Combat Environment Test and Evaluation Facility (ACETEF), Patuxent River, Maryland

- Developed the ACETEF federate software, integrated the new federate with the RTI, federate logger software and other JADS federation components
- Provided Phase 3 integration software support
- Provided Phase 3 test facilities and ACETEF test management personnel
- Provided the use of the Advanced Tactical Electronic Warfare Environment Simulator (ATEWES)
- Provided the anechoic chamber and personnel for aircraft support while in the chamber
- Provided Phase 3 network support
- Provided Phase 3 test support personnel

2.2.1.5 Georgia Tech Research Institute (GTRI), Atlanta, Georgia

- Provided test execution support at ACETEF
- Developed test control and execution methodology and automated capabilities
- Provide analysis and technical support for test data reduction and correlation
- Provided SME for V&V of the distributed environment
- Developed and integrated the ALQ-131 digibus monitor

2.2.1.6 Air National Guard Air Force Reserve Test Center (AATC), Tucson, Arizona

- Provided F-16 aircraft
- Provided ALQ-131 SPJ
- Provided aircraft and SPJ support personnel

2.2.1.7 Defense Modeling and Simulation Organization (DMSO), Alexandria, Virginia

- Provided RTI technical support
- Provided access to the RTI developers
- Delivered an RTI that met JADS latency and performance requirements

2.2.2 Assumptions and Constraints

During the test design phase, JADS developed the EW Test by applying a set of goals and constraints relative to test content, cost, schedule, and personnel described in the EW Test APA. The primary programmatic constraints were cost and schedule. In some ways, this test was a simple example of cost as an independent variable (CAIV). Technical content of the test was the primary area available for trade to maintain cost and schedule. Technical limitations (imposed because of cost and schedule constraints) caused a limited set of open air range instrumentation and constrained rules of engagement (ROE) to be used to support a single reference test condition (RTC) during the Phase 1 baseline data collection. Each of these technical constraints is discussed in the Content Constraints section of the Phase 1 TAP/DMAP. The Phase 3 test duplicated the same RTC used in the Phase 1 and Phase 2 tests. The impacts of the nontechnical constraints, cost, schedule, and personnel, are discussed in separate sections. The net result of the constraints is that the RTC and threat systems engagement represent a simple subset of developmental testing for an EW SPJ. This subset was sufficient for examining the impact of ADS on EW testing.

2.2.2.1 Cost

JADS had an established test budget and designed Phase 3 within the budget starting from original cost estimates. Because of funding limitations established for the EW Test, the resulting design represented the minimum ISTF test required to evaluate the utility of applying ADS to EW T&E. Although two straight weeks of testing were not the desired approach (this allowed very little time to assess or correct anomalies), based on the allotted timeframe coupled with AFEWES facility availability and the budget, this was the best, most affordable option.

Table 2. Phase 3 Cost Summary

Cost Item	Amount
AFEWES Support	$626,300
Georgia Tech Research Institute	$254,801
ACETEF	$775,500
Phase 3 Total	$1,656,601

2.2.2.2 Schedule

The primary Phase 3 schedule constraint was scheduling the overall JADS EW Test program. The JADS JTF charter has personnel assigned through the end of fiscal (FY) 99. The EW Test was designed for completion within the current JADS charter. Consequently, Phase 2 had to be completed by December 1998 so the Phase 3 test, scheduled in April 1999, and all reports could be completed before 1 October 1999. It was imperative that all major events required for Phase 3

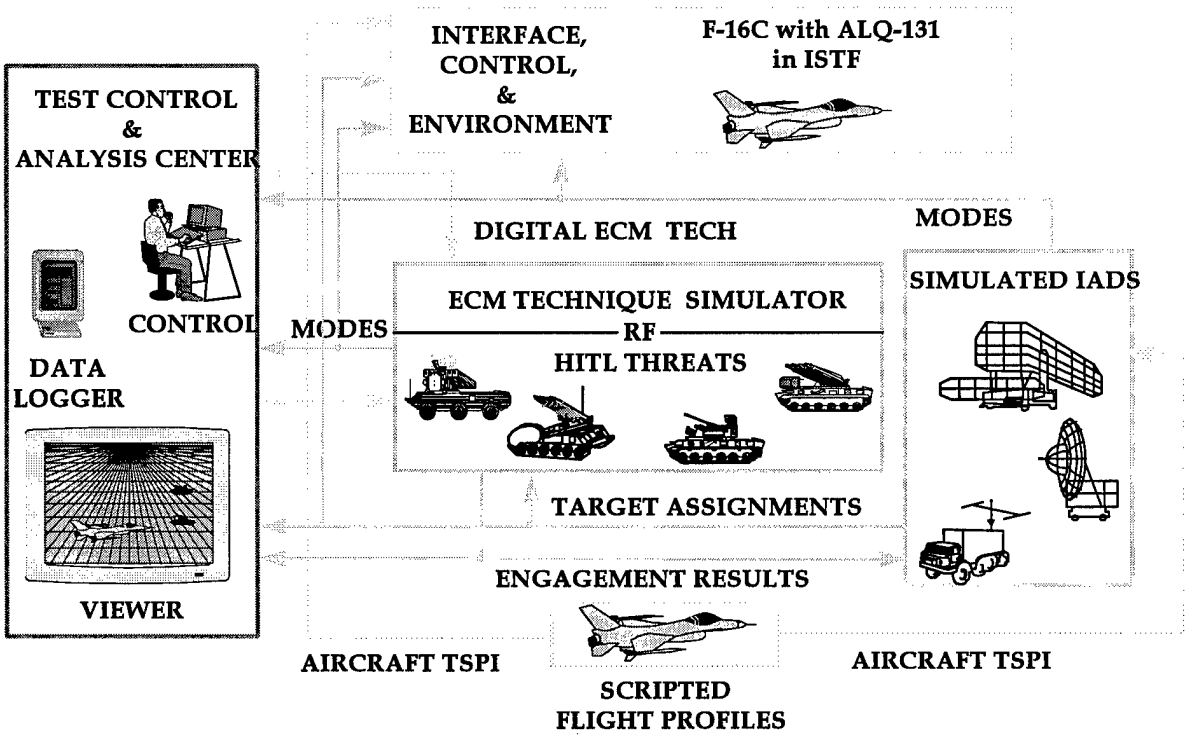
occur within a very limited test execution planning window. Thus, planning and preparation of Phase 3 test components overlapped with Phase 2 test activities.

2.2.2.3 Personnel

The final Phase 3 constraint was conducting the test program using current JTF assigned personnel augmented by experienced contractor and test facility personnel. Adequate personnel were available to complete the Phase 3 test. However, the plan included only two active threats at AFEWES because of their personnel limitations. Four manned threats were active for approximately three hours of the entire test period to provide some information on capabilities for expanded tests in the future. The test team compensated for this shortfall by incorporating the radio frequency environment (RFENV) federate to publish data from the unmanned threats.

2.3 Test Approach

The Phase 3 test approach changed only the SUT configuration from that used in Phase 2 testing. Instead of the digital system model of the SUT, Phase 3 tested the ALQ-131 receiver processor in two places: 1) hung on an F-16 installed in the ACETEF anechoic chamber, and 2) installed on a laboratory bench inside the ACETEF facility. The ACETEF SUT configuration was linked with HITL terminal threats at the AFEWES facility and a scripted model of the terminal threat hand-off portion of an integrated air defense system (IADS). The threat lay down from the OAR was replicated in the synthetic ADS-based environment and the SUT was flown, via the scripted flight profiles developed from the actual OAR flights and initial HITL test, against the AFEWES threats. This phase ultimately evaluated the ability to apply increased fidelity and resources through ADS late in the development cycle of a SPJ system through actual effectiveness testing of a proposed system before flight testing. The Phase 3 test configuration is illustrated in Figure 2.



TSPI = time-space-position information

Figure 2. Phase 3 Test Components

2.4 Test Objectives

The EW Test objectives assessed the utility of ADS. Significant technical challenges existed in implementing ADS in this environment as well as programmatic issues such as cost and schedule impacts. It is difficult at best to derive the utility assessment within the EW Test T&E framework without measurable objectives that provide insights for cost savings or value added. The overall EW Test objectives are outlined in the JADS EW Test APA and Program Level TAP/DMAP. Excerpts of the EW Test objectives that apply directly to Phase 3 are listed in Table 3.

Table 3. Phase 3 Test Objectives

Objective #	Objective
1-1	Establish SPJ performance in JADS/ISTF environment
1-2	Establish the repeatability of ISTF test results
1-3	Establish ranges of ISTF statistics for event data
1-4	Establish range of correlation coefficients for series observables
1-5	Quantify the effects of data latency on JADS/ISTF test environment
1-6	Quantify the operating reliability and mean time between failure of the JADS network
1-7	Determine the connectivity performance of the JADS network

2.5 Methodology

The Phase 3 test federation was designed to evaluate the utility of ADS for EW T&E. JADS executed two ADS test phases. The first ADS phase (Phase 2) utilized a software model of the ALQ-131 SPJ. The second ADS test phase (Phase 3) utilized the ALQ-131 SPJ pod mounted on an F-16 aircraft operating in an anechoic chamber.

The JADS EW Test methodology fully incorporated Department of Defense (DoD) high level architecture, which requires some description as to how it related to the Phase 2 and Phase 3 test methodology. HLA is an object-oriented approach to developing interactive simulation models and environments. The HLA consists of functional elements, interfaces, and design rules pertaining to interfacing simulation applications and intends to provide a common framework within which a specific architecture can be defined. For each simulation, an object model was built providing an appropriate abstraction of the objects, attributes, associations, and interactions used by the simulation. JADS EW Test used multiple simulations interacting to form what was called an HLA federation (for more information about HLA, see the HLA website at http://hla.dmso.mil/). This set of interacting simulations, or federates, along with their respective object models represents a federation object model (FOM). The JADS EW Test federation was used, with a supporting HLA RTI, to execute an ADS-based test representing the OAR Phase 1 test and range environment.

Two types of federates were used for the Phase 3 test: playback federates and pass-through federates. They are differentiated by their computer architecture and operational function. An example of a pass-through federate is the test control federate (TCF). The TCF had an Silicon Graphics, Inc., (SGI) O_2 computer hosting the UNIX-based RTI interface software linked to multiple personal computers (PCs) running the Automated Data Reduction Software (ADRS) application. The TCF software was written in C language under Windows NT. Pass-through federates were responsible for publishing data generated by application software and transmitting data subscribed from the JADS federation to the PC software application. The TCF was responsible for starting the federation execution of a trial and displaying some of the relevant test data needed for monitoring the test execution. The TCF federate was responsible for ensuring that the ADRS was supplied with all the necessary data to perform data reduction, analysis, and test visualization. The jammer federate was a pass-through federate responsible for passing data to and from the jammer to the rest of the federation.

Playback federates are designed to model an important OAR component by playing back a data script of key interactions or attributes recorded in Phase 1 test events - hence their designation as playback federates. Playback federates in the Phase 3 test were the platform, RF environment, and terminal threat hand-off (TTH) federates. Playback federates were responsible for publishing data elements from predefined scripts of data attributes and interactions to the JADS federation at correct times during the simulation of an OAR test run. The playback data were loaded during the joining process and transmitted based on a timed sequence of events. For each platform script generated, there were corresponding TTH and RFENV scripts for each active threat pair. During the ADS excursion runs when all four simulated threats were manned at AFEWES, no script was generated for or published by the RFENV federate.

2.5.1 Test Scenario

The F-16 aircraft, called the platform federate, replicated flying one simple profile at 360 knots, at an altitude of 9,000 feet mean sea level (msl), identical to the Phase 1 OAR flight profile. The AFEWES threat simulators representing the Simulated Air Defense System (SADS) III, SADS VI, SADS VIII, and the Weapon Evaluation Simulated Threat (WEST) X simulated threats, comparable to those used in Phase 1, were human operator controlled via scripted voice commands from the AFEWES test controller who was cued by TTH messages. AFEWES operated the threats in pairs—SADS VIII/WEST X and SADS III/SADS VI. These particular pairs were chosen to accommodate AFEWES manning considerations. To provide a controlled experiment evaluating the utility of ADS, a set of ROE was used during all phases of the JADS EW Test. These rules were intended to constrain the WTR operator actions to those that were easily repeatable and could be accomplished at the AFEWES facility while allowing some freedom to engage the aircraft. Additionally, because of the limited number of operators at AFEWES, all four threats could not be operated simultaneously without severe performance degradation. Therefore, RFENV transmitted the scripted modes of the inactive threat pair (SADS III and SADS VI) to the jammer coincident with the site controller matrix, while AFEWES operated the other pair (SADS VIII and WEST X). Once a sufficient amount of data was collected, the previous nonactive threats became active at AFEWES (SADS III and SADS VI), and the RFENV federate generated the modes for the initially active threats now inactive (SADS VIII and WEST X). JADS conducted both Phase 2 and Phase 3 testing this way because it was not feasible to activate all four threat systems simultaneously at AFEWES for the entire test period. The threat simulators engaged and disengaged the aircraft on each northbound and southbound flight profile which replicated the OAR test. Before execution began with the second pair of threats active, AFEWES conducted some engagements with all four threats active in their facility. The Phase 3 test runs where all threat systems were active were identified as ADS excursions. The ADS excursions were only used for ADS analysis and not for SPJ MOP calculations.

2.5.2 Rules of Engagement

The ROE for Phase 3 were driven by the requirement to rerun the OAR engagements using the Phase 3 test architecture. Descriptions of the ROE used are delineated in Appendix A of the Phase 1 TAP/DMAP.

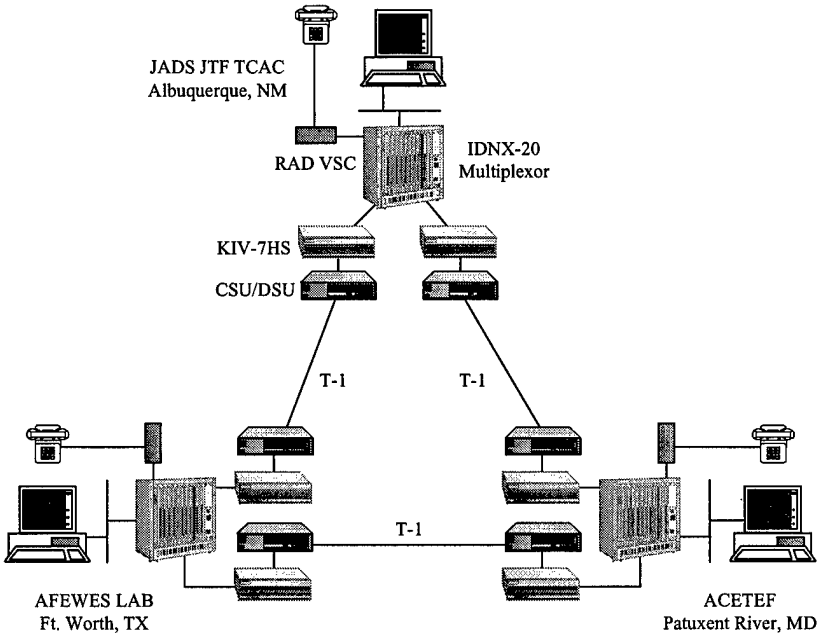
The ROE restricted the use of site operator enhancements such as electronic counter-countermeasures (ECCM), optics, and moving target indicator (MTI) modes. AFEWES operators used the same systems capabilities and operator techniques following the Phase 1 ROE. The EW Test team members closely monitored operator implementation of the ROE to maximize collection of useable data and limit variability induced by human actions. The aircraft flight path was generated from actual time-space-position information (TSPI) recorded in Phase 1. This flight path was designed to mitigate the background effects of clutter, glint, and multipath on the OAR.

2.5.3 Test Configuration

The JADS Network and Engineering (N&E) team and the EW Test team cooperatively developed the wide area network (WAN) and local are network (LAN) within the TCAC based on cost, resident knowledge, and software requirements. The same WAN and LAN (at JADS) were used for Phase 2 and Phase 3 tests. The federate software was developed and designed to function in the HLA communicating via the DMSO RTI. The EW Test team used JADS-purchased hardware and software to link the respective locations.

2.5.3.1 Wide Area Network Components

This section describes the equipment installed by the JADS JTF at each location (JADS, ACETEF, and AFEWES) comprising the WAN. Although it was not a requirement, it was desirable to utilize commercial-off-the-shelf (COTS) equipment that was easily obtainable and reasonably affordable. JADS JTF procured all the WAN equipment from existing government contracts with significant cost savings compared to the vendors' list prices.



CSU = channel service unit DSU = data service unit IDNX™ = Integrated Digital Network Exchange
KIV = AlliedSignal embedded KG-84 (a family of communications security equipment) communications security module
RAD = company that manufactures the voice signal converter
T-1 = digital carrier used to transmit a formatted digital signal at 1.544 megabits per second VSC= voice signal converter

Figure 3. Wide Area Network Components

2.5.3.1.1 Hubs

Originally, simple, unmanaged (dumb) Ethernet hubs were used to interconnect the various computer workstations and the router at a site into a single Ethernet segment. Prior to the Phase 2 test JADS JTF utilized a variety of unintelligent 10 megabits per second (Mbps) half-duplex Ethernet hubs that were available from multiple vendors. For reasons discussed below, JADS replaced the dumb hubs with Ethernet switched hubs.

2.5.3.1.2 Ethernet Switched Hubs

The Ethernet switched hubs were used in the same manner as the dumb hubs to interconnect the computer workstations and the router at a site to form a single Ethernet segment. The main difference between a switched hub and a dumb hub is that a switched hub will selectively route packets between ports, whereas a dumb hub will retransmit all incoming packets to all ports. On a switched hub, only broadcast packets are retransmitted to all ports. Switched hubs also operate at full duplex while dumb hubs operate at half duplex. Dumb hubs can have many Ethernet packet collisions while switched hubs show few, if any, collisions. The EW Test used SGI workstations running the IRIX operating system. The IRIX transmission control protocol (TCP)/Internet protocol (IP) stack had difficulty dealing with collisions. Collisions on the network would cause large latencies in reliable message traffic. This was avoided by implementing 10/100Base-T auto-sensing Ethernet switches that were available from multiple vendors. The auto-sensing feature of the Ethernet switches allowed JADS to connect 10 megabit (Mb) (e.g., the routers) and 100 Mb systems to the same Ethernet device.

2.5.3.1.3 Channel Service Unit/Data Service Unit

The channel service unit (CSU)/data service unit (DSU) interfaced the KIV-7HS encryption device or the Integrated Digital Network Exchange (IDNX™) trunk module to the T-1 communications line by converting the nonreturn to zero (NRZ) output of the KIV-7HS to a bipolar alternate mark inversion (AMI) signal for transmission over the telecommunications carrier facilities. In addition, the CSU/DSU supported binary eighth zero substitution (B8ZS) encoding and inserted framing bits in the extended super frame (ESF) format. Also, the model (VERILINK AS2000) of CSU/DSU used by the test networks was capable of remote configuration management and monitoring.

2.5.3.1.4 KIV-7HS Encryption Device

The KIV-7HS is a National Security Agency (NSA)-certified link encryption device that was used to protect the data being transferred between sites. The KIV-7HS protects classified and sensitive digital data transmissions (Type I) at data rates up to 1.544 Mbps. Its performance characteristics are similar with the KG series of cryptographic equipment. The KIV-7HS supports the T-1 data rate with one-way, end-to-end latency through a pair of KIV-7HS encryption devices of 4.5 microseconds. Also, the primary reason for using the KIV-7HS was the significant cost savings over the KG series of encryption devices. The cost of installing a pair of KIV-7HS encryption devices on a communications circuit was $7,969 versus $20,800 to install a pair of KG-194 encryption devices.

2.5.3.1.5 Integrated Digital Network Exchange

The Integrated Digital Network Exchange (IDNX™) is a communications resource manager (CRM) (multiplexer) that supports and integrates a broad range of voice, data, and internetworking services. The entire network was monitored, managed and controlled from any IDNX node in the network. JADS JTF chose the IDNX-20 series of CRM because of these features and the IDNX family of products was extensively used by the Defense Information Systems Agency (DISA) in support of the Defense Information Systems Network (DISN). The ability to configure and manage the systems from a single location allowed JADS to quickly troubleshoot problems and reconfigure the network equipment. The following subsections describe the feature modules utilized by the EW Test.

2.5.3.1.5.1 I422 Trunk Card

The I422 trunk card provided an RS-449/422 compatible interface for the IDNX to interface with the KIV-7HS or the CSU/DSU (nonsecure applications). The module also contained a cryptographic synchronization relay that allowed it to support automatic external resynchronization of encryption equipment. The I422 trunk module did real-time multiplexing, synchronization, internodal signaling, and contained the logic to control allocation of trunk channels. It allocated 16 kilobits per second (Kbps) of the T-1 bandwidth to an internodal communications channel that was the sole means by which nodes communicated with one another. The channel carried data that allowed the network manager to configure, query, and monitor all nodes from anywhere in the network. The internodal channel provided

- Call processing, configuration, network events, and status information to all nodes in the network
- Code loading when the desired code was not present in the node
- Database information, events, alarms, and circuit management messages to the network manager
- Continuous bit error rate test (BERT) in 30-minute intervals on the communications circuit

2.5.3.1.5.2 PX-3 and Access PX Router Modules

The packet exchange (PX) platform is a general-purpose router/bridge module integrated into the IDNX CRM. The PX platform provided packet-switched services among LANs over a WAN through the IDNX CRM. The module connected the TCAC LAN to the WAN via an Ethernet (Institute of Electrical and Electronics Engineers [IEEE] Standard 802.3). The PX platform featured an onboard processor and up to eight high-speed serial ports. PX platform serial ports can be connected to remote PX modules or to local or remote data cards with external serial ports. The PX-3 module was implemented for the EW Test because it supports IP multicasting. The PX-3 module utilized Cisco release 11.1 for its operating system. In addition, the PX-3 module was year 2000 (Y2K) compliant.

2.5.3.1.5.3 Quad Analog Voice Processor Module

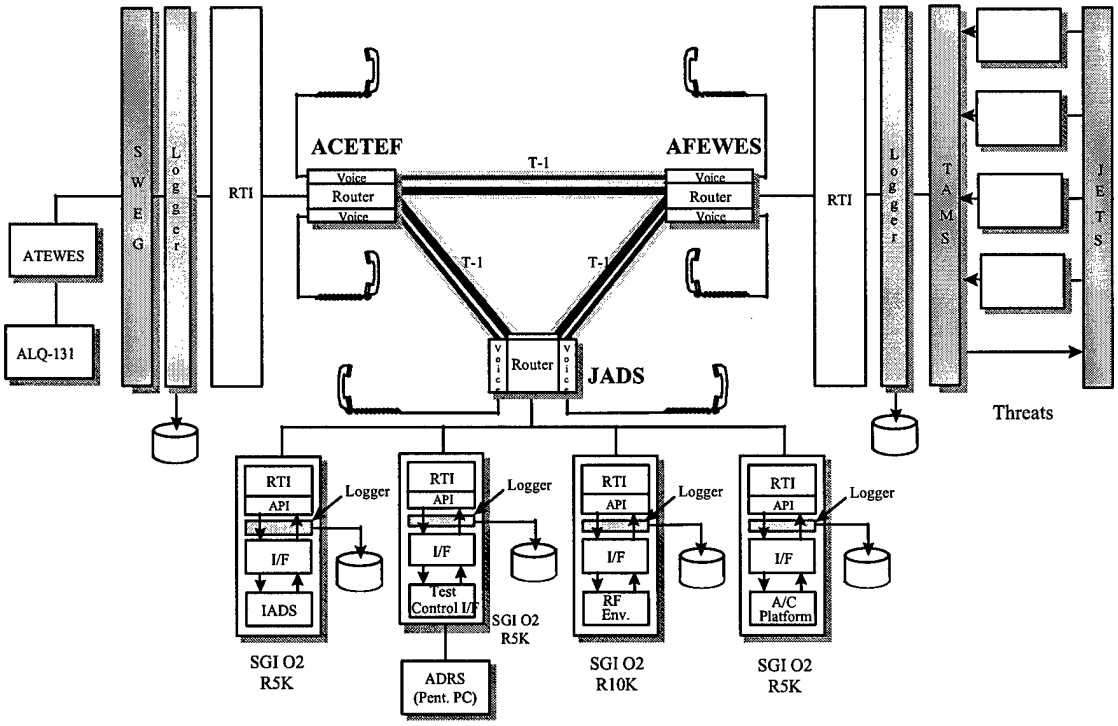
The quad analog voice processor (QAVP) module provided and managed voice calls coming into and leaving the WAN. It served as the interface between external voice communications equipment and the rest of the network. The QAVP module supported four full-duplex channels, which connected to industry standard four-wire E&M analog communications equipment. The module converted 3 kilohertz (kHz) bandwidth analog signals to 64 Kbps digital pulse code modulation (PCM) and vice versa. It featured echo cancellation, which eliminated echo caused by hybrid transformers that connected two-wire circuits with analog four-wire circuits.

2.5.3.1.5.4 RAD Voice Signal Converter

The RAD voice signal converter (VSC) interfaced between an ordinary two-wire telephone set and the four-wire E&M interface, enabling direct connection to the analog interface of a time division multiplexer. The VSC recognized the telephone set pulses for on hook, off hook and dialing; translated the pulses into the proper signaling standard; and sent the resulting signal over the "M" lead. When detecting activity on the "E" lead, the VSC sent the ring signal to the telephone and the ring back tone to the four-wire E&M interface of the QAVP.

2.5.3.2 Federate Components

The set of simulations and the model comprising the JADS EW Test Phase 3 interacted via the services of the HLA RTI in accordance with the JADS EW Test FOM and a common HLA rule set. To illustrate where the RTI resided, Figure 4 depicts the flow of information through the RTI within the entire federation as well as the individual federates.



A/C = aircraft API = application program interface I/F = interface pent = pentium
SWEG = Simulated Warfare Environment Generator

T-1 = digital carrier used to transmit a formatted digital signal at 1.544 megabits per second

Figure 4. JADS EW Test Federation

2.5.3.2.1 Description of the Jammer Federate at ACETEF

The SPJ used for the JADS EW Test was the ALQ-131 Block II jamming pod, an automatic, highly reliable, modular self-protection system. It was designed to provide advanced broadband coverage against red, blue, and gray radar-guided weapons. The ALQ-131 can be carried externally on a variety of front-line, high-performance aircraft. Internal installations are also available.

The modular design of the pod structure and electronic assemblies, plus its central computer software architecture, enabled the ALQ-131 system to adapt quickly to a broad spectrum of EW applications. The functional organization of the SPJ system centered on the interface and control (I/C) module containing a programmable digital computer as the system controller. The modules required for a given configuration connect to the I/C by a digibus that carries all sensor and control data. A memory loader/verifier allowed operational flight and mission specific program software to be loaded into the pod on the flight line in less than 15 minutes. The I/C module also contained a digital waveform generator that permitted broadcast of up to 48 simultaneous waveforms for deception modulation. When any ECM technique required a deception waveform, the values were transmitted to the onboard equipment via the waveform distribution bus.

The ALQ-131 Block II pod, pictured in Figure 5, included a receiver processor (R/P) which is a self-contained single modular package within the jamming pod. It enhanced the operation of this ECM system by maximizing its jamming capability and effectiveness against a multiple threat environment. This enhancement was accomplished through the concept of power management. The R/P was a wide-band, frequency-agile, double-conversion, super-heterodyne receiver using a crystal video receiver for low-band coverage. The module had a self-contained processor that performed automatic signal sorting and threat identification.

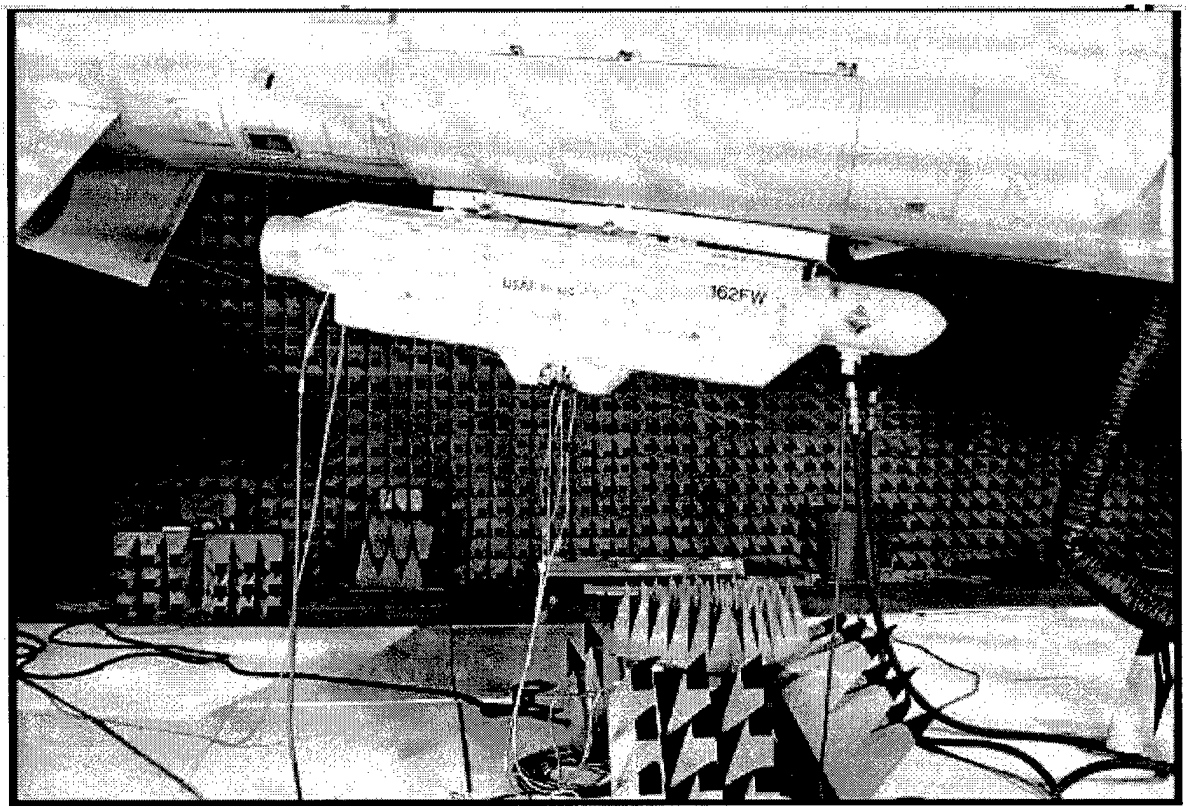


Figure 5. ALQ-131 Self-Protection Jammer Pod

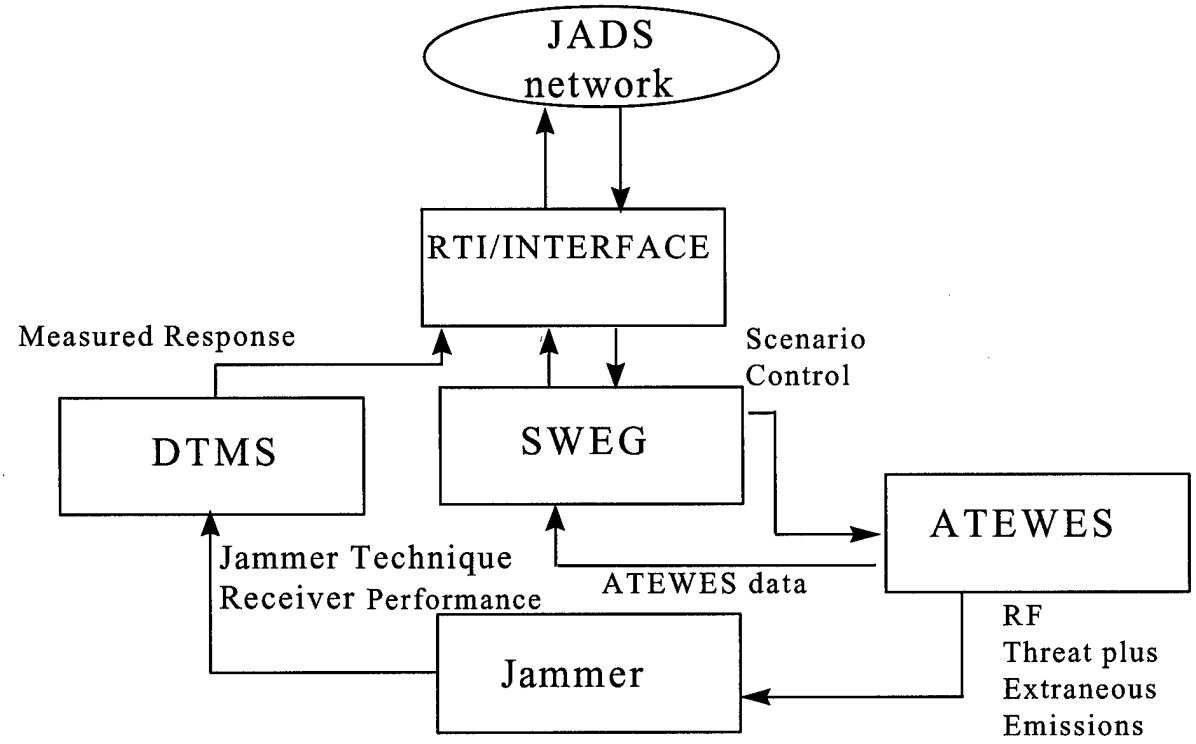
General operation of the ALQ-131 is as follows. The receiver conducts a signal search within a prescribed frequency range under the control of the processor. When a threat signal is acquired, the signal is analyzed for parameter values, formatted, and jammed if appropriate. The jamming energy is applied with optimum timing and is better concentrated with the emitter radio/intermediate frequencies and servo bandwidths. The ALQ-131 Block II pod was configured in Phase 3 with a bus interface, software for the pod operational flight program (OFP) and the R/P OFP. The pod also contained a limited threat simulator detection preflight message tape (PT) and a limited threat simulator response message tape (MT).

2.5.3.2.1.1 Description of the Digibus Traffic Monitor System

The ALQ-131 Digibus Traffic Monitor System (DTMS) was first used during Phase 1 in HITL tests with the SPJ at AFEWES. It was a PC-based digital instrumentation system built by GTRI. It monitored, displayed, and recorded pod traffic in real time. The DTMS was comprised of a PC running Windows NT, a Maxine card (printed wire assembly) that served as a digibus interface for the PC, a digibus traffic monitor (DTM) card (digibus traffic monitor printed wire assembly) used to record digibus traffic, an enhanced input/output (I/O) buffer module (EIOB) used to control and instrument the pod millicomputer in real time, interconnecting cables and assemblies, and specialized software. The DTMS was adaptable to specific test requirements and could be reconfigured. The DTMS provided instrumentation and monitoring of ALQ-131 operations in the ACETEF lab and inside the anechoic chamber during the Phase 3 test.

2.5.3.2.1.2 ACETEF Facility and Network

The components of the ACETEF architecture supporting the EW Test are shown in Figure 6. In Phase 3, the jammer was installed as part of the ADS test architecture. This configuration provided closed-loop effectiveness testing on an installed system using an ADS architecture.



Jammer - ALQ-131 Block II jamming pod SWEG - Simulation Warfare Environment Generator

Figure 6. ACETEF Phase 3 Network

Data items defined in the JADS EW Test ICD received or published by the jammer federate included

- Execution control messages
- Platform live entity state messages
- Multispectral (MS) source mode change messages
- Threat performance messages
- Link health check messages
- User-defined quality assurance (QA) data messages
- SUT jammer technique commanded messages

2.5.3.2.1.3 Universal Coordinated Time Code (UTC) Interface

The BanComm bc635PC provided an accurate time reference for the JADS federates. The time and frequency processing (TFP) card was connected to an external time source, a global positioning system (GPS) receiver, to provide a common time synchronization for the JADS federation. The TFP card used a Windows-based driver to maintain the PC system clock in synchronization with the time reference. This synchronization was transparent to the DTMS.

2.5.3.2.1.4 Simulated Warfare Environment Generator (SWEG)

The SWEG was an event-driven system that facilitated the interactions among various entities, virtual and real world. Virtually, everything existed within the SWEG scenario; however, any number of the SWEG 'players' could be controlled by real-world 'assets' with varying levels of detail.

In Phase 3 of the JADS EW Test, SWEG provided an avenue for multiple assets to interact with one another. The assets were Advanced Tactical Electronic Warfare Evaluation Simulator (ATEWES) emitter simulator, ATEWES log (dx files), tactical plot (TP) viewer, and HLA federations. All these entities connected through simulation warfare environment generator data (SWEDAT) which reflected a shared memory set up by SWEG to communicate to external assets and also provided them with the capability to communicate with one another. SWEG did this by allocating shared memory blocks and assigning 'mailboxes' in this shared memory to facilitate the passing of data among the various assets. SWEG also provided 'player structures' that the various assets could manipulate to display their interactions in a virtual environment.

2.5.3.2.1.5 Advanced Tactical Electronic Warfare Evaluation Simulator (ATEWES)

The ATEWES was an electronic warfare environment simulator capable of providing a three-dimensional electromagnetic environment at RF simulating up to 1024 emitters with up to four simultaneous pulses each microsecond on up to 255 moving platforms to an EW SUT. Frequency coverage for the ATEWES was continuous from 500 megahertz (MHz) to 18 gigahertz (GHz). ATEWES generated 1000 packets per second (Kpps) with no more than 2 percent dropped pulses at any single frequency distributed across 0.5-18.0 GHz at up to 250 Kpps. Maximum pulse density was 4 million packets per second (Mpps) and four simultaneous pulses per

microsecond. The RF sources contained nine digitally tuned phase settling synthesizers and six special channels for high duty cycle emitters. These sources were capable of

- 125 KHz frequency resolution with ±1 part per million frequency accuracy
- Pulse modulation on/off ratio of 70-decibel meter (dBm) minimum
- Pulse width (PW)/pulse repetition interval (PRI) resolution of 50 nanoseconds (ns)
- PW/PRI accuracy of ±20 ns
- Pulse amplitude resolution of 0.25 dBm with an attenuation range of 0-127.75 dBm

The ATEWES distributed RF signals via amplitude angle of arrival (AOA) distributions to provide the SUT sensors with correct angle of arrival and field of view stimulation. ATEWES supported 32 frequency-limited SUT receiver antenna patterns for each sector with 8192-bit samples covering 0 to 359.956 degrees, 0 to 63.75 dB attenuation range, and 0.25 dB attenuation resolution.

The ATEWES operated in stand-alone mode or integrated with the SWEG for external emitter control. The ATEWES also accepted other threat stimulator inputs, incorporated these signals into the dense emitter environment and distributed them to the SUT receivers via injection or radiation.

In support of JADS Phase 3 tests, ATEWES was used in the integrated mode to stimulate the ALQ-131 receiver processor system with RF signals whose parameters, tracking, and location are dictated by AFEWES. ATEWES replicated the threat modes of the closed-loop simulators at AFEWES. Jammer responses (1553 commanded) as well as the ATEWES RF parameters were sent across the network to AFEWES and JADS.

During the first week of Phase 3 testing, the ALQ-131 pod was located within the ACETEF laboratories and put into a bench configuration. The second week of testing was conducted with the pod installed on an aircraft suspended in the anechoic chamber (described in section 2.5.4.2.1). The ATEWES generated RF threats and applied them to the ALQ-131 receiver input via direct injection. ATEWES was programmed with a specific (although generic) ALQ-131 receiver antenna pattern for this test. This test simulated combined electromagnetic compatibility (EMC)/electromagnetic interference (EMI) and jammer effectiveness testing. Signal injection was used in place of free space radiation to isolate onboard EMC/EMI from threat induced effects.

2.5.3.2.1.6 ACETEF HLA/RTI Interface

The ACETEF HLA/RTI interface connected the RTI to several key components that make up the ACETEF jammer federate. It was the "hub" between the RTI and the digibus, SWEG, ATEWES, and the jammer. The interface was programmed to behave according to the specific federation in which it was a part. For the JADS EW Test federation, the work was specialized as follows.

1. The digibus was reached through a TCP/IP interface within the HLA interface. Digibus health, SUT jammer technique, and SUT perceptions originated with the digibus and were sent to the rest of the federation.

2. Incoming button discrete commands were routed to the digibus by the HLA interface.

3. The HLA interface continually broadcast its presence and link health check status to the rest of the federation.

4. The HLA interface received platform data, converted them into SWEG format (x, y, z) and passed the data on to SWEG. These geographical coordinates of the platform were essential to the proper positioning of the emitters with respect to the SUT in ATEWES.

5. Threat performance data (i.e., the tracking error of each of the four simulated threat land sites with respect to the platform) drove "ghost" aircraft in the SWEG environment. These ghost aircraft were entities only within the SWEG environment and had no relevance to the other federates. Each simulated threat emitter (in SWEG and ATEWES) tracked a particular ghost, whose SWEG x, y, z were computed from the platform's true x, y, z and the tracking error received in the threat performance data. Therefore, since each ATEWES emitter was tracking a ghost and not the actual platform, the power levels received by the SUT were lessened by the proper amount to reflect the actual tracking error.

6. The HLA interface received mode code changes and used them to affect the emitters in ATEWES. Whereas the threat performance drove the positioning of the emitter with respect to the SUT, mode codes drove the status of the emitter (on or off). Each of the four threats had one or more mode codes and one or more ATEWES emitters associated with it (not necessarily the same amount), so part of the task of the HLA interface was to accept the mode code data, decode which threat they originated from and which ATEWES emitter they represented, and pass along the status change to the proper emitter. The interface also kept track of the status of all the ATEWES emitters to accommodate messages received out of order, etc.

7. ATEWES itself responded to mode code changes by putting out a message in the form of QA data. These data were accepted by the HLA interface and passed to the RTI.

2.5.3.2.2 AFEWES Threats Federate

The AFEWES facility consisted of multiple man-in-the-loop simulations of threats, an internal LAN, sophisticated EW effects generators, diverse computer systems integrating AFEWES capabilities, test management capabilities, and a gateway for linking with external facilities. The external gateway was designed to support the ADS-based testing in Phase 2 and Phase 3. This was the only component of the AFEWES federate that did not exist prior to the JADS EW Test. These integrated components were used to support the test. This federate was responsible for providing the terminal threat simulations to the federation. These simulations included human-operated threat simulators designed to track a simulated target. The simulated jammer output from the target aircraft was injected as RF into each threat simulation to provide a realistic EW engagement. The threats were provided an RF simulation of the SPJ technique waveforms using the JETS. Figure 7 represents the facility components used during the Phase 3 test. The HLA

interface (gateway) within TAMS was the major application residing at AFEWES specifically developed for this test. It permitted communication among AFEWES and the other federates by using ICD-compliant message formats. The logger was another piece of software used. It collected specific federate data processed both in and out of the interface. The AFEWES federate required no other software or hardware additions to facilitate test execution.

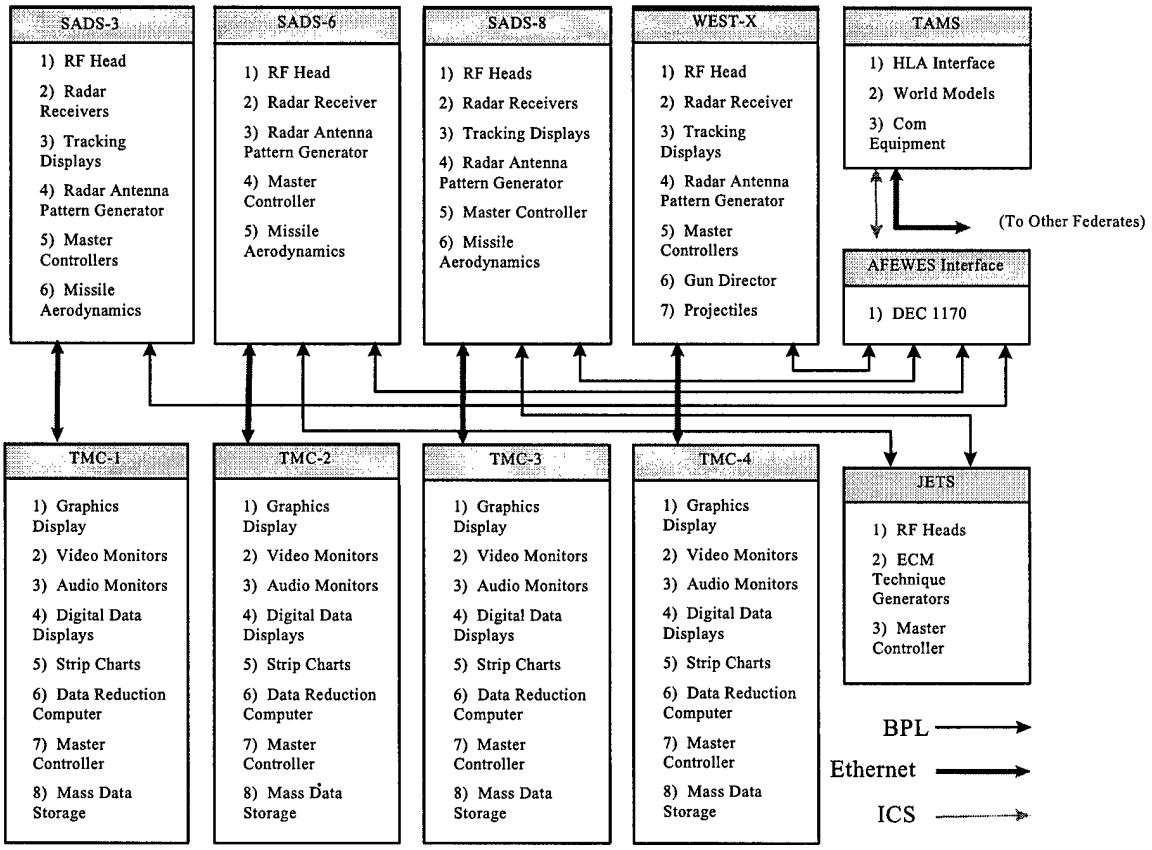


Figure 7. AFEWES Federate Configuration

2.5.3.2.2.1 Threats

The AFEWES closed-loop, surface-to-air weapon system simulations for this test consisted of an RF head; a tracking console; a software-programmable antenna pattern generator (SPAG) computer and customized software programs; and a master computer, which provided overall real-time control of the target tracking simulation. The RF head simulated real-time echo signals. It also modulated and scaled RF signals to simulate the effects of range, antenna gain patterns, and other factors in the radar range equation. The tracking console contained receivers, target tracking servo-control systems, a system synchronizer, simulation displays, and man-in-the-loop radar operator controls. Antenna patterns were simulated on the SPAG computer through a table look-up process. The simulators used during JADS tests were the SADS III, SADS VI, SADS VIII, and WEST X. The JADS and ACETEF federates interfaced to the AFEWES simulators through the TAMS computer which hosted the AFEWES threats federate that provided an HLA-

compliant interface to AFEWES and each simulator. The TAMS computer was an SGI Challenge multiprocessor system. A Digital Equipment Corporation (DEC) 11/70 was used as an interface buffer among the four simulators and the TAMS. The TAMS data represented the aircraft platform consisting of TSPI, attitude, radar cross section (RCS), and the pod antenna patterns. The AFEWES federate in TAMS took raw simulator output data and formatted them according to the JADS federation ICD. It also converted input data from other federates into a format readable by the appropriate simulator.

2.5.3.2.2.2 Test Management Centers

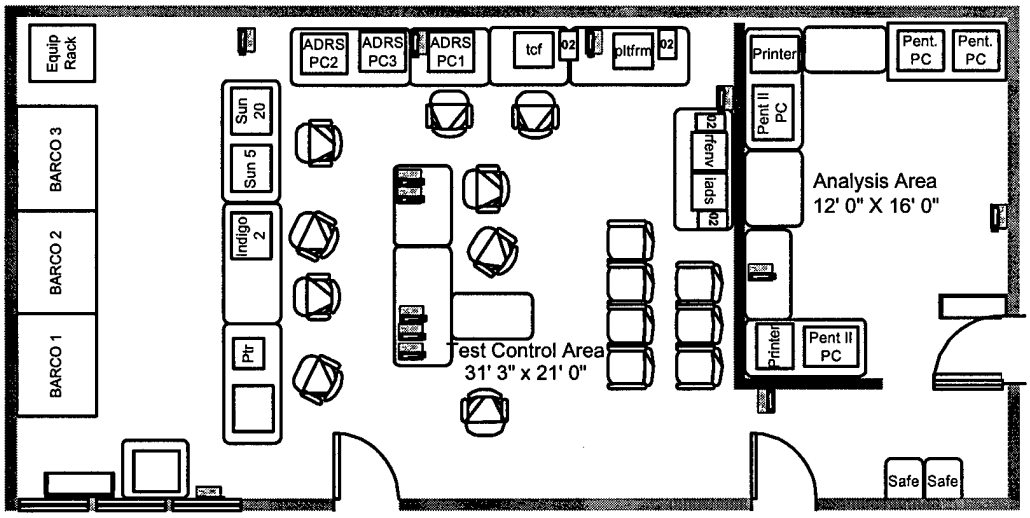
The AFEWES test management centers (TMC) monitored and collected data such as jamming-to-signal ratio (J/S), radar tracking error, and missile/projectile miss distance from the four closed-loop simulators. A separate TMC was associated with each simulator. Raw data collected during the test were available for evaluation via digital strip chart printouts, graphics, miss distance calculations, and radarscope video tapes.

2.5.3.2.2.3 JammEr Technique Simulator (JETS)

The AFEWES JETS was used to generate SUT ECM techniques to AFEWES simulators for the Phase 3 test. Specifically, during the test runs involving generation of closed-loop engagements with simulations of the SADS VIII and the WEST X, JETS supplied RF responses representing the SUT emissions. Then during the generation of closed-loop simulations of the SADS III and SADS VI M, JETS supplied representative SUT ECM technique emissions.

2.5.3.2.3 TCAC Test Federates

In addition to the federates located at AFEWES and the DSM located at ACETEF described previously, the five other federates located at JADS in the TCAC (see Figure 8) were the platform, RF environment, terminal threat hand-off, analysis, and test control federates.



ADRS Automated Data Reduction System used for test control and post-test data reduction

Barco	Large screen display systems used to display various computer screens
Equip Rack	Equipment rack with Barco switches, GPS receiver, tape recorders, phones, etc.
IADS	Terminal threat hand-off federate
indigo2	SGI Indigo2 used as a network and engineering workstation
O_2	SGI O_2 workstation
Pent PC	Pentium PCs used for post-test analysis
Pent II PC	Pentium II PCs used for post-test analysis
pltfrm	Platform federate
RFENV	RF environment platform
TCF	Test control federate
sun20	Sun SparcStation20 used to host *SPECTRUM*, a network monitoring tool
sun5	Sun SparcStation5 used to host the analysis federate from TRAC Monterey

Figure 8. EW Test Control and Analysis Center

2.5.3.2.3.1 Test Control Federate (TCF)

The TCF managed the test execution and collection of necessary data to evaluate SPJ and ADS performance measures. The TCF interfaced with the ADRS computers as a pass-through federate to propagate the setup, start-up, and stop commands as well as performing other test control and display functions. The ADRS setup command initiated loading the corresponding run scripts in the platform, TTH and RFENV playback federates located in the TCAC. TCF transmitted the setup, start-up, and stop commands to all federates (DSM, AFEWES, RFENV, TTH, platform, and analysis) to control the beginning and ending of the run. The TCF federate also provided the required HLA interface capability for the PCs hosting the ADRS applications. TCF passed relevant status, performance, and position data for other federates at JADS, AFEWES, and ACETEF to ADRS, which provided unique EW Test visualization during each test run. Visualization features included real-time displays of jammer and threat emitter status, a radar warning receiver (RWR), and a heads-up display (HUD) showing aircraft attitude, altitude, and speed. Specific EW performance displays (e.g., J/S ratio and tracking error) were also provided by the ADRS during each run. A total of three PCs hosting ADRS were connected to the TCF. Two of the ADRS PCs were used for real-time monitoring by JADS analysts during each run. The third ADRS PC provided a "hot spare" for the analysts in case the ADRS software crashed during a run.

2.5.3.2.3.2 Platform Federate

The F-16 aircraft flight path modeled in Phase 3 (called the platform federate) provided a composite of position and attitude. No systems or characteristics of the F-16 were represented in the model. Aircraft position and attitude for each test run recorded from the OAR TSPI and inertial navigation system (INS) were played back in the form of a data script by the platform federate. For one threat, the platform federate script also played back threat-centric tracking error measured in the OAR. Since AFEWES operated SADS VIM and the WTR had a SADS VI, the data recorded on the range were used to set up the engagement at AFEWES. In order to ensure proper sequencing of SADS VI tracking data with aircraft position, threat performance data for the SADS VI target track radar (TTR) were also included in this script. The TSPI position and attitude as a function of time corresponded to a real pass made in the OAR to have maximum value for correlation. Each threat simulator at AFEWES responded to an RF signal that represented a theoretical reflection of the aircraft. The signal was built at AFEWES using the aircraft position and attitude in relation to the threat and "looking up" an RCS value from a table.

A four-way power interpolation was performed based on these table values to obtain the corrected RCS values for the exact aircraft position relative to the site. The signal was then modified to account for its distance and motion relative to the threat as well as the relationship between the threat antenna boresight and the aircraft position. This was also the way the F-16 RCS was presented during the HITL test.

2.5.3.2.3.3 RF Environment (RFENV) Federate

This play-back federate was responsible for reporting emissions of the two unmanned simulators to the jammer federate as collected on the range. The data were time ordered (as they occurred on the range) and replayed based on an elapsed time from start sent by ADRS through the TCF. In order to replicate the test environment from the OAR test phase during the ADS phases, these extraneous emissions were simulated in the ADS test environment as modes for injection into the DSM/jammer federate. RFENV was not programmed to transmit data when AFEWES was operating all four threats. Originally, the RFENV federate concept was not designed for this purpose. It was designed to emulate the extraneous RF emissions and signal distortions recorded during the OAR test. These signals were to be applied to the ADS environment through this federate. Unfortunately, the instrumentation used to record and subsequently recreate the conditions of the OAR test was inadequate for this requirement. Since the data were not available for replication, RFENV evolved into a federate populating the environment—published data for an unmanned threat pair—to maintain the appropriate threat density for the SPJ.

2.5.3.2.3.4 Terminal Threat Hand-Off (TTH) Federate

This federate was responsible for assigning terminal threats located at AFEWES to the target simulated by the platform federate during an engagement. The test controller at AFEWES received a visual cue on the TAMS to turn on/off the appropriate threats. The test controller transmitted a voice command relaying the on/off cues to the threats sent from the TTH federate. The time for each cue was taken from the OAR site controller matrix. This information was scripted and played back sequentially based on elapsed time from the start command. The initial intent of this federate was to act as the command and control element of the threats by transmitting digital commands directly to the simulators; however, AFEWES was unable to support this technical design without modifications to some simulations. The alternative implementation used digital commands sent from the TCAC to the AFEWES gateway via the TTH federate, and then the test controller read the commands off the display to the threat operators. This approach more closely duplicated the voice command structure used during the OAR and HITL phases.

2.5.3.2.3.5 Analysis Federate

The analysis federate had many of the same functions as ADRS; however, it took a different approach to data collection and scenario visualization. It was another useful visualization tool for the test controller. It produced a top-down view of the scenario similar to ADRS but also showed threat site modes and missile flyouts on the same map display. Other displays on the perimeter showed the EW MOPS and measures of effectiveness (MOE) which were calculated as

the scenario evolved. It added the benefit of EW Test MOP analysis in real time, and also aided in quality assurance of the data and troubleshooting of anomalies.

2.5.4 Instrumentation

JADS used various types of instrumentation for the Phase 3 test described below.

2.5.4.1 TrueTime Global Positioning System Receiver

The GPS receiver was a time source provided by the GPS satellite constellation. It had Inter-Range Instrumentation Group - Format B (IRIG-B), 1 MHz, 5 MHz, and 10 MHz signal outputs for use by timing distribution systems.

2.5.4.2 BanComm Timing Cards

The computers in the TCAC and at ACETEF all contained BanComm cards connected to the IRIG-B time code signal from a GPS receiver. All the federate software and the DSM software executing on the DSM PC obtained time directly from the BanComm cards. The PCs running ADRS also contained BanComm cards. However, because the ADRS software was a Windows 16-bit application but BanComm only had 32-bit drivers, those PCs obtained their time from the PC system time, which was periodically synchronized to GPS time by a BanComm utility program.

2.5.4.3 JADS RTI Interface Logger

The logger resided in the software interface between the federate and the RTI. It recorded all function calls to and from the RTI along with all the function data parameters. For example, when the federate wanted to publish data, it called the RTI updateAttributeValues function. When the logger was linked with the federate, the federate called the logger updateAttributeValues function. The logger stored the function identification and parameter data in the log file buffer and then called the RTI updateAttributeValues function. When a log file buffer became full, it was written asynchronously to the log file and a new buffer was created.

The logger was designed to minimize impact on the federate with which it was linked. To accomplish this, the logger design included the following features: asynchronous direct I/O, nondegrading process priority, and binary file format.

Asynchronous I/O was used so that the federate software did not wait while the data were written to the log file. When a buffer became full, an I/O request was queued to the operating system and control was immediately returned to the federate. A separate process accomplished the actual writing of the data.

Direct I/O allowed the operating system to use the data buffer created by the logger software to write the data to the disk. Normally, the operating system copied the data from the user buffer to a system buffer. However, use of direct I/O eliminated this copy operation.

If a user with super-user privileges executed the logger within the federate, the logger would take advantage of system nondegrading priorities to further minimize the impact of the logger I/O on the federate. When asynchronous I/O was initialized, a set of processes was created to perform the writing of the data to disk. When these processes were created, they inherited the priority of the process that created them. The logger software lowered the priority of the process before the asynchronous I/O was initialized. After the I/O processes were created, the logger software set the federate process priority to a real-time priority. Since the I/O processes executed a lower priority than the federate process, the I/O processes never interfered with the federate process.

The log file created by this software was a binary file. Attribute and interaction data were received by the logger (or by the federate) in a binary format. In the interest of minimizing the processing time used by the logger, the binary data received from or sent to the RTI were written directly to the log file without any conversion.

Since the logger software writes all the binary data sent to or received from the RTI without attempting to translate or convert them, the logger can be linked with any federation without modifications to the logger software. Also, since the logger classes were derived from the RTI base classes, very few lines of code must be changed to incorporate the logger into an existing federate. Less than twenty lines of code were modified or added to link the logger with the helloWorld demo program provided with the RTI.

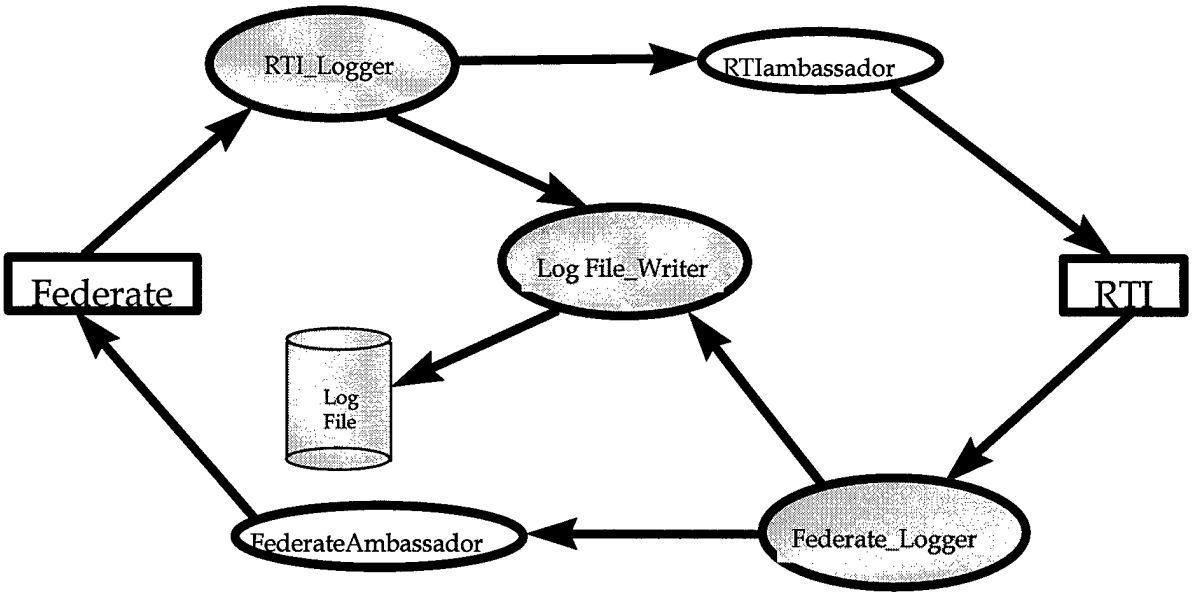


Figure 9. HLA Logger Implementation Diagram

2.5.4.4 Network Monitoring

A combination of in-house tools and commercially developed software products provided JADS with a real-time, or near real-time, limited capability to assess network performance and evaluate

the integrity of data as they were being collected during Phase 3 testing. The various tools were used to help provide a clear picture of the network and to speed diagnostic and maintenance efforts during a test. Data on the network were not collected for post-test analysis during Phase 2. As is discussed below, this was changed for Phase 3 based upon observations made during Phase 2 post-test analysis.

Simple Network Management Protocol (SNMP) tools were used to monitor communications and network hardware. This allowed JADS personnel to see, in near real time, the status of the long-haul links as well as the routers connecting remote sites. The SNMP tools also allowed JADS to monitor and record the bandwidth used on the T-1 links.

Each federate sent link health check updates and displayed the information received from other federates. This was used during Phase 3 testing not only to determine the health of the each federate but as another view of the overall health of the network.

In addition, a simple utility used standard pings to display the status of various federation computers. This tool often presented the test team and networking personnel with the first indication that a problem existed with the network and/or the computers at each site.

For some runs, data dropouts and high latency spikes were noticed in post-test analysis. Router statistics were examined in real time using a very intrusive processor to determine if the routers were the problem. There was no indication that the routers were dropping packets and latency could not be examined with the available data. It was determined that protocol analyzers, or sniffers, located on each LAN segment at each node allowed JADS personnel to determine the causes of some of the data dropouts and high latencies. Sniffers were not used during Phase 2 because changes to the Ethernet architecture would have been necessary at each site and the required numbers of sniffers were not available. Sniffers were incorporated into the Phase 3 architecture.

2.5.4.4.1 Network Traffic Analysis

SPECTRUM®, a network analysis package developed by Cabletron Systems, provided a near real-time capability for network traffic monitoring presenting current packet rate and load information, as well as packet error and discard rate information for network equipment. In addition, *SPECTRUM* Alarm Manager, with simple diagnostic capability, was valuable in the detection and troubleshooting of network outages. *SPECTRUM* utilized the SNMP to periodically query network devices and display requested information on screen in table and graph format. The *SPECTRUM* operator tailored the destination, frequency, and content of the queries to provide the desired level of insight into a particular network portion or piece of equipment. Typically, a thirty-second polling frequency was used to monitor EW Test equipment. *SPECTRUM* event log and query results were also stored to database for post-test analysis.

2.5.4.4.2 Link Availability Monitor

There were numerous ways JADS personnel gained insight into the availability of a particular EW Test network link. For instance, a sudden drop in packet rate picked up by *SPECTRUM* usually

indicated a network link problem. Another solution made use of the self-diagnostic capabilities of the network equipment. A line printer in the TCAC was set up to print diagnostic messages directly from the IDNX multiplexer. The sound of the printer in motion drew immediate attention to a potential equipment outage. JADS programmers coded another simple tool, based on the UNIX *ping* utility, that allowed test controllers to quickly verify link availability with a glance at one screen. Called the "stop light" tool, it presented a small green, yellow, or red on-screen graphic for each monitored link based on the current status. If pings were delayed or dropped, the status of the link changed. Ping data were not stored for future analysis.

2.5.4.5 Network Health Check

For the Phase 3 test, there were two types of periodic federate health checks. The test showed that neither type provided completely satisfactory instrumentation for that purpose.

2.5.4.5.1 RTI Heartbeat

The first health check was the internal RTI "heartbeat" message sent every six seconds via TCP/IP from each federate to the federation executive (FEDEX). If the FEDEX failed to detect three successive heartbeat messages from a federate, then it would display a warning message in the FEDEX window on the SGI O_2 hosting the RFENV federate.

2.5.4.5.2 Federate Link Health Check.

The federate "link health check" (LHC) system, as documented in the Phase 3 ICD, was the second health check used during the test. This system employed 1 hertz (Hz) LHC messages sent best effort from every federate to every other federate. It proved to be much more useful than the heartbeat, both in real time during the test runs and later during the post-test analysis, because of its higher frequency and the fact that the JADS RTI logger captured the LHC messages.

2.5.5 Test Control and Monitoring

2.5.5.1 Test Control and Analysis Center (TCAC)

The TCAC located at JADS served as the hub for test control and data collection for Phase 3 tests. The five federates (platform, TCF, TTH, analysis, and RFENV) resided in the TCAC and the network monitoring, data collection and storage, test visualization and analysis were also performed from within the TCAC. The TCAC test controller and operators had voice communications to the two sites, AFEWES and ACETEF, and were able to relay federation commands and speak directly with the site observers over a conference phone system. The TCAC systems provided the test manager with the capability to monitor the test and control the execution with the assistance of the site observers and the other federate operators.

2.5.5.2 Site Observers

JADS representatives were positioned at AFEWES and ACETEF to observe critical test elements or events. These observers used on-site visualization tools as well as direct observation of operator actions to provide additional insight during test execution and post-test analysis. Observers at AFEWES provided detailed notes of simulated threat actions, JETS operations and the AFEWES federate. For each run, the observers noted whether the run was considered usable or unusable for analysis based on the appropriate responses from a particular system. All visible anomalies were noted too. This information was helpful and necessary in discerning the quality of usable data during the analysis process. Although detailed information regarding the engagement was readily available at AFEWES in the form of strip charts, the handwritten notes augmented these digital printouts considerably.

2.5.6 Runtime Infrastructure Software

The JADS federation implemented the RTI Version 1.3R5 for SGI O_2 computers using the IRIX 6.3 operating system. The federates conformed to Version 1.3 of the HLA interface specification.

2.5.7 JADS EW Federation Object Model (FOM)

In simple terms, a FOM is the identification of the objects, to include their attributes and interactions, used in a specific federation. The FOM used by the JADS EW Test was prepared in accordance with the HLA Object Model Template, Version 1.3. The JADS federation implemented for the Phase 3 test was the JADS EW Test FOM Version 1.0.

2.5.8 JADS EW Test Interface Control Document (ICD)

The JADS EW Test ICD specified the HLA interface requirements among JADS EW Test federation members at a level sufficient to implement all requisite RTI service calls. The ICD was developed to augment the FOM because information required to develop and implement the JADS EW Test federation was lacking from the object model template. The ICD version used for the JADS EW Test Phase 3 was Version 1.5, dated February 18, 1999. The ICD and FOM were critical to implementing the HLA and executing the JADS Phase 3 test.

2.6 Schedule

The schedule shown in Table 4 outlines the major tasks and associated execution time windows. This matrix includes the preliminary actions required prior to the Phase 3 test.

Table 4. Test Event Schedule

Event	Start Date	Completion Date
DTMS integration test at ACETEF	1 Mar 99	9 Apr 99
Sling available at ACETEF	15 Mar 99	30 Mar 99
ALQ-131 arrives at ACETEF	30 Mar 99	2 Apr 99
F-16 arrives at ACETEF	14 Apr 99	15 Apr 99
F-16 into chamber	16 Apr 99	16 Apr 99
Phase 3 test readiness review (TRR)	8 Apr 99	8 Apr 99
Execute federation integration tests	12 Apr 99	13 Apr 99
Preliminary Phase 3 testing w/o F-16	14 Apr 99	16 Apr 99
Execute Phase 3 w/F-16	19 Apr 99	23Apr 99
Quick-look report (daily)	14 Apr 99	23 Apr 99
Quick-look report (summary)	30 Apr 99	N/A

2.7 Security

The highest classification level of data processed by the EW Test team was secret/US only. The highest level of data reported was secret. To ensure the proper classification of data collected and presented, the EW Test team incorporated classification standard operating procedures and information security policy from elements directly related to the test. Additionally, the SPJ security guide was also available for use.

2.7.1 Network Security

The Phase 3 test established both secure voice and secure digital by using KIV-7 encryption devices throughout the WAN. These devices permitted point-to-point transfer and verbal transmissions of classified information. Once JADS reached security agreements with ACETEF and AFEWES, the WAN segments were operationally ready to handle classified data.

2.7.2 Data Security

JADS signed formal security agreements with each facility to pass information up to the security classification of secret across the network. However, most of the data transmitted from node-to-node were unclassified. Data exchange, safeguarding and labeling were commensurate with security classification guides and policies established by the governing agencies.

3.0 Preliminary Testing Events

The development of the required components for the EW Test, the integration of simulators, software, and test facilities, and the implementation of the network architecture were an incremental process completed in a series of many steps. Preparations began when the JADS EW Test was chartered in August 1996. The following April the EW test approach was baselined to include the use of the emerging HLA rather than the established distributed interactive simulation (DIS) protocols. In December 1997, JADS built a network test bed and began the first of many test activities focused on the computer, communications, and HLA RTI software supporting the ADS-based test phases. JADS worked closely with DMSO during the development of software (e.g., RTI versions) and tools for federation documentation (e.g., federation execution planner's workbook, object model developer's tool kit). JADS took initial delivery of key federate software components from GTRI in August 1998 and began stand-alone test and integration of the federate software. The final software components were delivered in November 1998 and acceptance testing was completed on all the federates comprising the Phase 2 test. Since all federates (except the DSM) were reused in Phase 3, much of the preliminary testing done for Phase 2 was not required again for Phase 3. This section describes the preliminary events supporting Phase 3 development.

3.1 Phase 3 Development Tasks

To ensure JADS was fully prepared and ready to accomplish the Phase 3 test, tasks with specific completion requirements and schedules were identified as risk reduction steps preceding the Phase 3 test readiness review milestone. These tasks are listed below and described in the following sections.

- Network testing (ACETEF only)
- RTI 1.3r5 performance assessment
- Integration of jammer federate
- ACETEF - jammer federate stand-alone acceptance testing
- Federation integration testing

3.2 Network Testing

JADS performed extensive network testing starting many months before the first ADS test phase. Before testing the RTI software in Phase 2, JADS wanted to characterize the network in the simplest form. To determine the raw network throughput performance, JADS developed software to send data one way from one computer to another. Versions of this software perform tests using TCP and IP with multicast data transport modes. The one-way software was designed to exercise the network with different data packet sizes and transmission rates. A complete matrix of rate and size combinations was tested. Each test case, defined by a specific rate and size pair, was conducted for a thirty-second duration. The one-way raw network test consisted of two programs – a sender and a receiver. At the start of each test case, the sender transmitted a "start" message to the receiver indicating the size, rate and total count of messages to be sent. The

receiver used this information to name the output file and to determine if any messages were lost. After sending the control message, the sender transmitted the data. The data packet contained a sequential serial number and the time (i.e., when it was time tagged in the sending code) the message was sent. When a message arrived at the receiver, the system time was obtained. To eliminate its effect on the latency calculation, no I/O occurred while the data were transmitted. The data file contained a record for each message that should have been received. If the message was received, the serial number, sent time, received time, and latency were written to the file. This sequence of steps was repeated for every combination of size and rate. This benchmark testing of the overall network was not repeated prior to Phase 3 since network components were the same as Phase 2.

3.2.1 Test Bed Development

The RTI test hardware configurations progressively increased in complexity until the entire federation and network architecture (except for the T-1 lines) was in place in the JADS test bed. Starting with a two computer point-to-point configuration shown in Figure 10, JADS N&E gathered basic performance data for network IP multicast data and network TCP data. Software testing was performed on the following RTI software and data types: RTI 1.0-2 best effort data, RTI 1.0-2 reliable data, RTI 1.3 beta (1.3b) best effort data, RTI 1.3b reliable data, RTI 1.3-2 early access version (RTI 1.3-2EAV) reliable data, and RTI 1.3-2 (early official release) reliable data.

The test configuration included all network components using a two-node network for the same series of tests. The associated communications link and hardware/software configuration were also tested. All sources of possible latency were computed through a disciplined process of adjusting one variable at a time and collecting recorded time data for the same message type transaction in differing reference test conditions. The two-node network test used an SGI O_2 5000 and an SGI O_2 10000 running IRIX 6.3. The test software and RTI were hosted on each computer for all tests using this configuration. Figure 10 shows the test bed configuration.

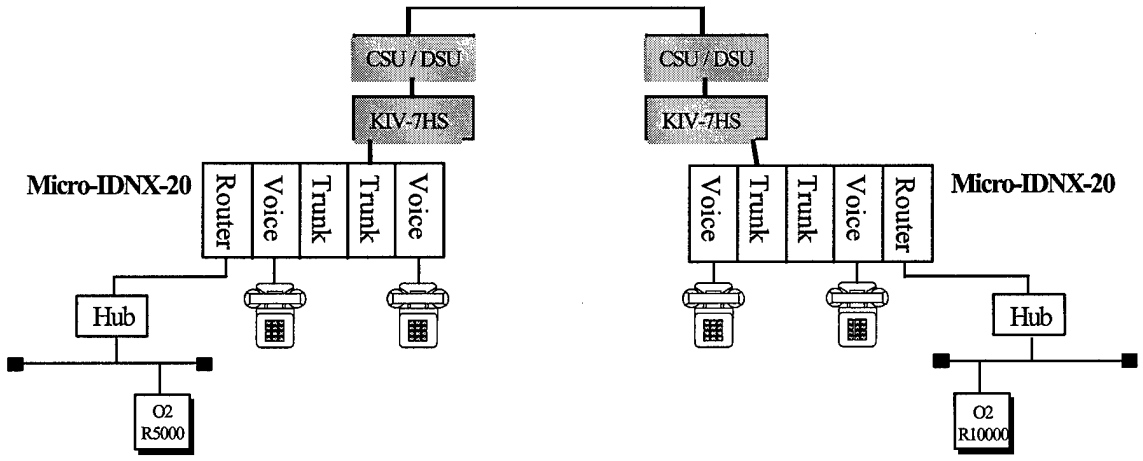


Figure 10. JADS 2-Node Test Bed Configuration with Communications Devices

3.3 RTI Performance Assessment

After characterizing the network in the simple one-way tests, JADS N&E needed to determine if the RTI would support the anticipated loads placed on it by the JADS federation in Phase 2 and Phase 3. The testfed federate, additional software developed to simulate RTI loading, accepted command line arguments that specified the characteristics of an instance of the federate. The user specified the federate identification (ID) number (-f), the duration of the test (-d), the size of the attributes and interactions (-s), the rate that attributes are published (-r), the number of updates at the specified rate (-n), the amount of time the federate should wait before starting to publish at its specified rate (-w), and whether interactions should be published (-i). There was only one attribute and one interaction to which all federates subscribed. JADS conducted the test by running with three federates residing on separate computers. For the three-federate test, the testfed was configured on one computer to publish 11 attribute updates at 20 Hz (simulating the AFEWES node). Another instance of testfed was configured to publish two attribute updates at 20 Hz (simulating the federates in the JADS TCAC node). The third instance of testfed was configured to publish one attribute update at 20 Hz (simulating the ACETEF node). All three federates published interactions at approximately 1 Hz. The size of attributes and interactions was 121 bytes. Attributes were published best effort and interactions were published reliable. The test team executed multiple tests with a duration between two and five minutes. After the three-federate tests, six-federate tests were run using six computers that resembled the Phase 2 configuration more closely. The tests identified some problems. These problems were fed back to the DMSO technical support for analysis. At the same time, JADS network and analysis personnel also analyzed the problems. In some cases the problems were in the network and/or federation configuration. In these cases, DMSO provided recommendations to correct the problem. In the cases where problems were in the RTI, fixes were implemented in subsequent RTI releases. As a new version of the RTI was released, JADS personnel exercised the RTI with the testfed software. Through this process, JADS learned invaluable information about using the RTI, provided feedback on problems and improvements to the developers, and ultimately gained confidence that the RTI would support Phase 2 and Phase 3 performance requirements. Table 5 lists the versions of the RTI tested by JADS. RTI 1.3-4 was used for the Phase 2 test execution. However, it had a problem of losing aircraft TSPI data if the federates did not join slowly in a specific sequence. This problem was fixed in RTI 1.3-5 and building on the iterative RTI testing and the Phase 2 test experience, JADS used this RTI version for Phase 3.

Table 5. RTI Versions Tested by JADS

RTI Version	Date Released
1.0-2	February 1998
1.3b	3 April 1998
1.3-2 EAV	15 May 1998
1.3-2	15 June 1998
1.3-4	October 1998
1.3-5	16 December 1998

The test environment expanded from the two-node configuration and used at least three and as many as six SGI O_2 workstations (either R5000 or R10000 models) running IRIX 6.3. The three-node test configuration in the EW Test test bed with three SGI computers is shown in Figure 11. Once the RTI performance baseline was verified, further testing, integration, and tuning of all Phase 3 federation components was performed.

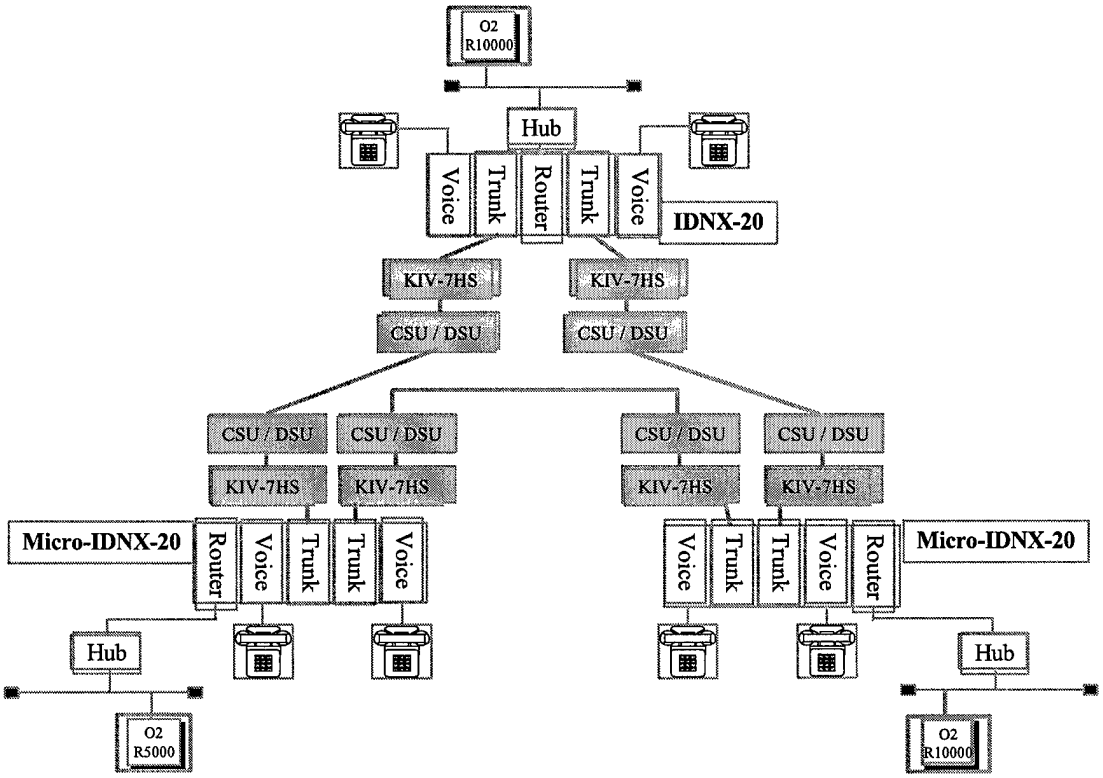


Figure 11. JADS 3-Node Test Bed Configuration

3.4 Phase 3 Integration

The integration of the hardware, software, and facility unique components (e.g., AFEWES HITL simulators, gateways) in preparation for Phase 3 testing was divided into several tasks intended to incrementally assemble and test the Phase 3 architecture. The major tasks representing Phase 3 integration were jammer federate acceptance testing (FAT), and federation integration testing (FIT).

3.5 Jammer Federate Acceptance Testing

Acceptance test procedures were developed to demonstrate that the jammer federate developed for the JADS EW Test Phase 3 was an adequate integration of the ALQ-131 Block II SPJ pod in the ACETEF ISTF. The ACETEF gateway federate was expected to accept inputs from the JADS federation and allow other equipment within ACETEF to convert those inputs into RF

waveforms, stimulate the SPJ pod, collect the outputs from the pod, and return those outputs to the federation. The outputs were used to allow JADS to study how ADS affects test results. As such, key characteristics of the federates had to match the characteristics of the systems or environments they simulated to avoid confusing differences between the federates and those systems/environments with ADS-induced differences. Because correct, timely functioning of these federates was critical to the EW Test, JADS determined that a formal acceptance test was necessary.

As such, key characteristics of the real ALQ-131 performance characteristics integrated at ACETEF were examined and verified to avoid confusing differences between the Phase 2 DSM and the 131 used on the OAR with differences induced by ADS. Because the jammer performance was so critical to the EW Test, JADS determined that this formal acceptance test was necessary. In addition, this FAT served as the verification, validation, and certification (VV&C) test for the ACETEF federate and all the systems within their facility necessary for stimulating and monitoring the pod.

The FAT consisted of four test phases. The first phase included a series of formal events that verified readiness of all component subsystems for the actual FAT tests. The second phase performed simple tests on each ACETEF (jammer) federate. The third phase tested each federate in a dynamic environment in which the HLA RTI and other messages are exchanged. In the fourth and final phase, the testing verified repeatable, robust, and proper functional performance in a dynamic environment in which the ACETEF-developed federate exchanged messages, some of which were sensitive to latency. During the last three phases, JADS collected sample data pertinent to various ADS measures. The four phases and the related functions addressed are listed below.

FAT Phase 1 - Formal Test Readiness Events
- FAT Phase 1 checklist
- Configuration certification of ACETEF-controlled subsystems
- Configuration management
- EW Test network and computer configurations
- HLA configuration
- Correct functioning of timing system
- RTI loggers and log file readers
- Traceability of common database files
- Traceability of database file contents
- FAT script selection
- FAT script generation, verification, and installation
- ADS measures

FAT Phase 2 - Simple Federate Tests
- FAT Phase 2 checklists
- FAT Phase 2 federate interoperability test
- RTI executive start-up
- RTI executive normal operation

Federation join/resign
Object discovery
Computer health display
Link health display
FAT Phase 2 exit criteria

FAT Phase 3 - Dynamic Federate Tests
FAT Phase 3 ACETEF federate tests
Federate publish/subscribe intentions
Actual federate publication/subscription
Federate status and error messages
Federate/facility integration tests
Subscription corruption
Federate/ATEWES integration
ATEWES/pod integration
Pod/digibus monitor integration
Digibus monitor/federate integration
Publication corruption
FAT Phase 3 AFEWES federate interoperability tests
Federate publish/subscribe intentions
AFEWES federate link health
AFEWES federate status and error messages
Federate/facility integration tests
Subscription corruption
Federate/ATEWES integration
ATEWES/pod integration
Pod/digibus monitor integration
Digibus monitor/federate integration
Publication corruption
FAT Phase 3 exit criteria

FAT Phase 4 - Full Complexity Tests
FAT Phase 4 scripts

3.5.1 Jammer Federate Acceptance Test Results

The test verified and validated the ACETEF federate prior to evaluation of the entire federation in the FIT. The test was completed on 12 April 1999. Execution was scheduled to be completed on 7 April 1999. On that date, overall FAT results depicted that phases 1 through 3 were complete. Phase 4 was extended to resolve other EW Test federation issues, not specifically the jammer federate. Results were briefed to the JADS EW Test accreditation panel on 8 April 1999. They recommended accreditation. Details on problems identified during the FAT follow.

- ACETEF attempted to migrate to a different processor than originally planned. This migration would have allowed ACETEF to quickly change to a different processor should the

HLA gateway computer fail. This attempt delayed the completion of FAT Phase 1 by half a day. When sufficient progress could not be made, JADS directed that the migration be abandoned. The HLA gateway ran on its originally planned processor.

- ACETEF, GTRI, and AATC diagnosed and removed a source of pod false alarms. GTRI also modified the DTMS software to correctly output status in case future false alarms were encountered. This code was tested prior to the removal of the false alarm source.
- GTRI corrected a message output from the DTMS that was not ICD compliant. This message was found in the examination of the messages published by ACETEF and was confirmed when the federation executed with AFEWES.
- During two runs, the ATEWES generated a signal to turn on the SADS VI illumination mode without being commanded to do so by the federation. One instance was attributed to a failure of ATEWES monitoring software being used to time tag ATEWES pulse commands.
- Phase 4 of the FAT was held open until 12 April because of an early missile launch from the SADS VI on southbound runs. This was suspected to be a problem at AFEWES. ACETEF was allowed to stand down on 9 April while AFEWES continued to work the problem. Testing on 12 April confirmed the AFEWES fix and the FAT was closed.
- Analysis Summary. Data were collected on both ADS and EW measures. Only latency and data loss were examined beyond real-time test monitoring. Data collection, transfer, and archival were exercised and accomplished as planned.
- ADS Measures. Examination of latency and data dropouts indicated the federation performed much better than in Phase 2 testing. This was attributed to a replacement of the router cards with Y2K compliant versions that had more memory, changing to RTI 1.3 release 5, and changes to the RTI initialization data (RID) file.

3.6 Federation Integration Test (FIT)

The FIT was conducted at JADS, ACETEF and AFEWES on 13 April 1999. Based on the previous results of the jammer FAT, experience gained from Phase 2 testing, and jammer integration tests at ACETEF, an abbreviated FIT provided the necessary results to conduct the Phase 3 test. The FIT was completed in approximately four hours. A predetermined number of Phase 3 northbound and southbound test runs focused primarily on the functionality, stability, and adequacy of the integrated software capability were conducted. The FIT results were used to verify the integration and performance of the complete Phase 3 test federation and architecture before any runs for record were made. The FIT addressed federation capabilities in the following areas:

- Phase 1 - Network components and time synchronization verification
- Phase 2 - Federation components and functionality
- Phase 3 - Test control and monitoring capabilities
- Phase 4 - Federation execution with manned threat pairs for northbound and southbound runs (wet and dry)
- Phase 5 - Data collection, retrieval, and analysis capabilities

3.6.1 FIT Results

During the FIT, all test start-up, execution, and stop procedures were verified. Site coordination and status control were exercised. Federate components (e.g., threat simulators, ATEWES, DTMS, etc.) and federation execution were fully demonstrated and verified. ACETEF federate hardware and software were fully exercised, and integration with AFEWES and JADS federate software was proven. All threats were demonstrated in pairs (SADS III, SADS VI and SADS VIII, WEST X) using northbound and southbound scenarios and all four threats simultaneously. AFEWES internal quality control data were compared to the Phase 1 HITL test run data to verify repeatability. Post-test data processing and analysis capability were performed and verified. Full analysis was not possible because the ADRS software was not calculating all MOPs, but this did not impact test execution readiness.

3.7 Verification and Validation

The JADS accreditation board previously accredited the Phase 2 test environment addressing the ADS architecture, federate functionality, and the federation fidelity as a whole. The Phase 3 test environment was the same as Phase 2 except for the jammer federate as mentioned previously. Consequently, for Phase 3, the V&V addressed critical functionality and fidelity aspects of the jammer in the ISTF. V&V addressed three key data sources: threat parameters, jammer parameters, and aircraft RCS. Accreditation of the threats consisted of a documentation search and key personnel interviews to determine limitations that could impact the JADS effort. During V&V for Phase 2, JADS was able to find limited accreditation information on the AFEWES threats. Accreditation information was directed at a threat baseline established through the intelligence community and documented in the Electronic Warfare Integrated Reprogramming (EWIR) Database. VV&C were provided by AFEWES following established OAR quality control procedures. On the receive side of the jammer, threat parameters from AFEWES were transformed by ATEWES RF signal generation capabilities (e.g., correct power, correct RF waveform). VV&C of threat parameters were performed prior to the FAT by GTRI and approved by the JADS accreditation board. Because of the critical functions ATEWES performed, continuous monitoring of threat parameters was done during Phase 3 execution. Jammer parameters and antenna patterns were addressed previously during Phase 2 VV&C. The jammer waveform output at AFEWES was thoroughly demonstrated in Phase 2 and verified by GTRI (at ACETEF) during the jammer FAT and again by AFEWES for Phase 3. The EW Test integrated product team (IPT) approved aircraft radar cross section and the VV&C were provided by the results of AFEWES quality control procedures. No issues were identified with key data sources during Phase 3 V&V. The results of these V&V activities were reported to and approved by the JADS accreditation board.

4.0 Test Execution and Control

Following the successful completion of development, component testing, acceptance, integration, and verification activities, the EW Test team performed a test readiness review at JADS. Based upon the work accomplished and the results achieved, the Phase 3 test was approved for execution. This section reports on the procedures used for test execution and provides the execution details about the Phase 3 test events.

4.1 Test Control

Test control is an important aspect of any formal test environment. To ensure that test conditions and status were monitored and maintained in a distributed, ADS-based test environment, JADS had to carefully define its requirements. Areas addressed included test control, unique real-time status reporting and display capabilities, and test control procedures within the TCAC as well as at AFEWES and ACETEF. Critical components of the Phase 3 test control included federation time synchronization, federation start-up, and status monitoring described below.

4.1.1 Federation Time Synchronization

All computers for the Phase 3 test used a common time source to ensure valid time values were recorded in the logs at all sites during test execution. Prior to the start of daily testing, a time synchronization test was run by the TCAC to verify that all computers were using the correct, accurate time source (IRIG-B) in their operations and not running on local internal system time. All JADS federates except AFEWES and the analysis federate participated in the time synchronization test. Since the analysis federate only read and recorded data, it recorded time-synchronized data from other federates. Time synchronization was performed by initiating a normal run and after a jamming response was observed in the TCAC the execution was stopped. JADS copied the TCF log file to the appropriate log file directory and ran the log file summary program on the file. The recorded times for all messages in federate log files were checked to look reasonable (within +/- 20 milliseconds of one another). The times recorded on the link health check messages showed the time on all the SGI O_2s. Execution control messages (start and stop) showed the time for the ADRS 2 PC. The time recorded on the X_File_Spec message showed the time for the ADRS 1 PC. The time recorded on the SUT messages showed the time for the jammer federate.

The AFEWES computers were also synchronized to an IRIG-B time source. Software executed on the AFEWES computers synchronized the system time to the IRIG-B time. The AFEWES federate software used the system time as its time source. To determine if AFEWES was synchronized, JADS would run the raw TCP test program previously used for RTI testing. Upon completion, the receiver printed out the minimum, maximum, and mean latency. JADS verified that the latency was reasonable. A real-time analysis capability was not available to determine if all systems maintained time synchronization during test runs. However, during subsequent data analysis the quality of time synchronization was able to be determined.

4.1.2 Federation Start-Up

During the integration tests for Phase 2, it was determined there were less data dropouts if the platform federate was not the first federate to join the federation. Since five federates in the TCAC were all subscribing to most of the same reliable data, DMSO technical support recommended that JADS execute with a single reliable distributor for the TCAC. Normally, each federate contained a reliable distributor to send and receive reliable data. There was a TCP connection from every reliable distributor to every other one. By configuring the TCAC federates to use one reliable distributor, the number of TCP connections and the subsequent traffic on the WAN was reduced. The RFENV federate did not have much data to process, so it was chosen as the host location for the TCAC reliable distributor (RELDISTR). The federate containing the reliable distributor started the federation executive (FEDEX) and joined the federation first. It was observed that if the FEDEX was not started on the same computer where the RTI executive (RTIEXEC) was running, it would take a few minutes for the FEDEX to find the RTIEXEC.

After the RFENV federate started and joined the federation, all of the other federates in the TCAC would join (staggered by a second or two). After the TCAC federates successfully joined the federation, the remote federates (AFEWES and jammer) joined.

4.1.3 Federate Status Monitoring

The link health check display as well as the FEDEX window was monitored during federation execution. If there were problems, generally one of these windows displayed an indication. For example, the link health check status display on the TCF screen indicated federate status as red or green. It provided the capability to monitor specific multicast traffic paths in one direction only between any two federation nodes (e.g., JADS - AFEWES). The FEDEX window monitored the RTI "heartbeat" messages over the reliable (TCP) data paths. The federate displayed red if the federate stopped sending link health messages because of software error or a failure in the link. However, the problem was generally noted by one of the remote sites before the TCAC federates exhibited a corresponding display. This was due to the fact that the TCAC had fewer indications of information outages from other nodes. The link health and FEDEX windows could only detect outages that occurred over an extended period of time (> 3 seconds for link health, 20 seconds for FEDEX). Status monitoring procedures of individual federates and operator procedures are described below in further detail.

4.1.3.1 Jammer (ACETEF) Operation

For Phase 3, the facilities at ACETEF hosted the ALQ-131 jammer. The ACETEF jammer federate operation involved two major activities for each test run.

The first activity managed the local processes to successfully join the federation and begin accepting data. This complex procedure consisted of several steps. The first step was to run a script that started the link health check monitor, the emitter monitor, a graphic visualizer called "TacPlot," SWEG, and the RTI and SWEG interfaces which comprised the ACETEF HLA interface. Immediately, SWEG would begin polling for external assets and continued to do so

until all external assets were ready. The external assets consisted of the emitters being controlled by ATEWES, the HLA interface, SWEG or TacPlot graphics, and the asset-to-asset communications. The RTI interface immediately polled the user for the run number, set up a shared memory connection with the SWEG interface, started the JADS logger file, joined the FEDEX, published and subscribed to all expected data, and finally rested in a pattern of polling the RTI for a start command (issued by TCAC) and broadcasting the status of the federate. The SWEG interface meanwhile established its shared memory connection to the RTI interface and waited for the start command issued by the RTI interface.

The ACETEF federate operator then vocally authorized the ATEWES operator to start, and once ATEWES came on line, SWEG recognized this and passed to its next waiting step. SWEG status was then waiting for a start execution order from the SWEG interface.

The ACETEF federate operator informed the JADS TCAC that ACETEF was ready for the start. Another good visual queue that ACETEF was ready for the start command was when the SWEG graphics screen was displayed.

JADS then sent the start command from the TCAC, which was received at the RTI interface and passed to the SWEG interface, which, in turn, then sent SWEG the start execution command. At this point, SWEG graphics would appear and the federate and the federation entered run mode.

The operator monitored the integrity of the federation with the tools available at ACETEF. This included the link health check monitor, the emitter monitor, SWEG or TacPlot graphics, and warning or dropout message information displayed on screen by the RTI interface. The link health check monitor indicated federate status and link check between federation nodes by red or green coloring. The emitter monitor displayed the different emitters used in the scenario and red or green coloring indicated their respective modes of operation. Warning or dropout messages were displayed to indicate missing and out of sequence data. If necessary, dead reckoning smoothed lost TSPI data. SWEG graphics or TacPlot were used for test execution control.

At the end of the run, the ACETEF federate operator shut down the link health check and emitter monitors, the RTI and SWEG interfaces, TacPlot, and SWEG. The ATEWES operator shut down the ATEWES.

At the end of each test day, the ACETEF federate operator collected all log files and organized them into permanent directories. The ACETEF federate logger file, the ATEWES timing file and the ATEWES dx file were all collected. These files consumed many megabytes of storage.

4.1.3.2 Test Control Federate (TCF) Operation

The TCF, located in the TCAC, started by executing the *tcf.sh* shell script. When the federate started, it prompted the operator for a run number. After the operator entered a run number, the federate prompted the operator to join the federation. When directed by the JADS test controller, the operator entered a 'y' at the prompt so that the federate joined the federation. When the TCF completely joined the federation (as indicated by "continue execution" displayed in the federate

window, the ADRS operators started the ADRS software on each of the three ADRS PCs. The software on each PC started a few seconds apart from one another to help prevent the TCF federate from periodically crashing. Once the ADRS software was running, the federate waited for execution control commands from ADRS. When an execution control attribute was received, the TCF federate published it for the other federates. The script names and the start and stop messages were sent to all the federates using this method. When the command was an execution control attribute with an execution control word that indicated stop test execution, the TCF federate published the attribute and then resigned from the federation.

4.1.3.3 Platform Federate Operation

The platform federate located in the TCAC started by executing the *platform.sh* shell script. When the federate started, it prompted the operator for a run number. After the operator entered a run number, the federate prompted the operator to join the federation. When directed by the JADS test controller, the operator entered a 'y' at the prompt so that the federate joined the federation. The platform federate waited until it received an X_File_Spec attribute update that contained the name of the script to be loaded. When it received the script name, the federate displayed the name of the script being loaded. Upon completion of script loading, the federate displayed "done." The federate then waited for an execution control attribute update with an execution control word that indicated the start of test execution.

During a federation execution, the platform federate executed without operator intervention. The window in which the federate executed was monitored for error messages. The platform federate played its script until an execution control attribute update was received with an execution control word that indicated stop of test execution. The federate then resigned from the federation.

4.1.3.4 Radio Frequency Environment (RFENV) Federate Operation

The RFENV federate located in the TCAC started first because it created the federation execution (FEDEX) and the reliable distributor (RELDISTR) used by all federates in the TCAC. The RFENV federate started by executing the *rfenv.sh* shell script. When the federate started, it prompted the operator for a run number. After the operator entered a run number, the RFENV federate created the FEDEX. The federate then prompted the operator to join the federation. The operator entered a 'y' at the prompt so that the federate joined the federation. The RFENV federate waited until it received an X_File_Spec attribute update that contained the name of the script to be loaded. When it received the script name, the federate displayed the name of the script being loaded. Upon completion of script loading, the federate displayed "done." The federate waited for an execution control attribute update with an execution control word that indicated start of test execution.

During a federation execution, the RFENV federate ran without operator intervention. The operator monitored the window in which the federate executed for error messages. The FEDEX window was also monitored for error messages indicating loss of contact with other federates. The RFENV played its script until an execution control attribute update was received with an execution control word that indicated the stop of test execution. The RFENV federate waited for

all other federates to resign from the federation and then resigned from and destroyed the federation execution. If there were problems with any federate resigning, the federation was destroyed manually by entering a "kill" command in the FEDEX window.

4.1.3.5 Terminal Threat Hand-Off (TTH) Federate Operation

The TTH federate located in the TCAC started by executing the *handoff.sh* shell script. When the federate started, it prompted the operator for a run number. After the operator entered a run number, the federate prompted the operator to join the federation. When directed by the JADS test controller, the operator entered a 'y' at the prompt so that the federate joined the federation. The TTH federate waited until it received an X_File_Spec attribute update containing the name of the script to be loaded. When it received the script name, the federate displayed the name of the script being loaded. Upon completion of script loading, the federate displayed "done." The federate then waited for an execution control attribute update with an execution control word indicating the start of test execution.

During a federation execution, the TTH federate executed without operator intervention. The window in which the federate executed was monitored for error messages. The federate played its script until an execution control attribute update was received with an execution control word that indicated the stop of test execution. The TTH federate then resigned from the federation.

4.1.3.6 AFEWES Threats Federate Operation

The AFEWES threats federate consisted of federate software hosted on an SGI computer and facility unique systems and software for scenario status control and display, test management centers, and operator consoles. AFEWES controlled the test run execution and individual systems from a central facility linked internally by intercoms with external voice links to JADS and ACETEF. The test controller at JADS advised the AFEWES controller by voice for federate and run start and stop conditions similar to the ACETEF federate. The AFEWES controller then coordinated internal execution actions with operators and advised JADS of current status.

4.1.3.7 Analysis Federate Operation

The analysis federate provided an improved scenario viewer for observing northbound and southbound test runs and specific threat engagements. It showed the specific modes a threat site used and missile flyouts as they occurred. Real-time displays of the 10 EW MOPS and the real-time values of jamming-to-signal ratio and tracking error were provided for situational awareness of each threat engagement with the target aircraft. Data collection and storage for the analysis federate were nearly automatic and required little operator intervention once the run started. Once the federation began, the analysis federate operator waited to join the federation. Queued from the TCF operator, the analysis federate joined the federation and awaited the start of the test execution. During the run, the window was monitored for errors in the threat performance or variance in the threat operators. Once the test execution completed, the operator resigned the analysis federate from the federation and immediately restarted the software to begin the next run.

4.2 Test Execution

JADS achieved all of the test execution and data collection objectives established for Phase 3. The EW Test test team executed the nine days of testing and exceeded the minimum goal of 62 runs per threat pair coupled with the several ADS excursion runs. This section reports on the test execution issues JADS experienced in conducting the Phase 3 test using an ADS environment. Causes of problems and analysis of the impacts are provided if they were identified during the conduct of the test. Much of the cause and effect examination of anomalies experienced was not accomplished until after the test during the detailed examination of all the collected data by JADS analysts. Where problems are identified in this section without a specific resolution given, they are addressed in the detailed post-test analysis (see Section 5).

The Phase 3 test exit criteria in Table 6 were all met. The successful test effort produced valuable data on EW and network performance which are provided later in this report. The ADS MOP results and evaluation are discussed in Section 5 of this report.

Table 6. Phase 3 Exit Criteria

Objective #	Phase 3 Exit Criteria
1-1	Complete at least 62 runs per threat pairs - twice (62x2 = 124 data samples)
1-2	Complete several ADS excursion runs
1-3	Establish ranges of jammer statistics for event data
1-4	Establish range of correlation coefficients for series observables
1-5	Quantify the effects of data latency on JADS/ISTF test environment
1-6	Quantify the operating reliability and mean time between failure of the JADS network
1-7	Determine the connectivity performance of the JADS network

4.2.1 Phase 3 Test Summaries

The test execution of each run followed a standard procedure. Through several risk reduction events, the test team determined that a stable test environment with the federates required a sequential, methodical joining process. Initially, the test began with the following sequence: RFENV federate, TTH federate, platform federate, TCF, ADRS, ACETEF jammer federate, AFEWES federate, and analysis federate. Although ADRS was not a federate, it provided the start and stop commands to the federates for each run. ADRS also sent the setup command to identify and load the appropriate scripts in the platform, RFENV, and TTH federates. Once the federates joined and the AFEWES federate objects were declared, the test controller gave the command to begin the run. The runs lasted for approximately three-and-a-half minutes. By the fifth day of testing, the run turn-around time (from the start of a run to the start of the subsequent run) was approximately seven minutes. The procedures were streamlined as much as feasibly possible. If critical data elements during the execution of a run were lost, federates crashed or connectivity was lost, simulators crashed or did not have a seemingly valid engagement, the run was terminated and not considered usable for analysis. Additionally, if an engagement was not

scored as good by AFEWES (threat simulator engagement), it was also counted as unusable for jammer MOP analysis but used for ADS assessment.

The JADS EW Test Phase 3 was conducted from 13 -23 April 1999. During the first week (13 - 16 April), test runs were conducted with AFEWES SADS VIII and WEST X threat simulators manned during northbound and southbound runs. The jammer federate was configured using the ALQ-131 not installed on an aircraft. The WEST X was not activated during southbound runs in order to duplicate the Phase 1 engagement scenario. Scripts representing the other two threats (SADS III and SADS VI) engagement activity against the jammer were generated by the RFENV federate. During the second week (19 - 23 April), test runs were made with AFEWES SADS III and SADS VI threat simulators manned during northbound and southbound runs. The jammer federate was configured using the ALQ-131 installed on the F-16 hanging in the anechoic chamber. Daily testing issues and results are summarized in Table 7 for all test runs.

Table 7. Phase 3 Test Execution Summary

	Total Runs	Max band width used	Peak round-trip latency	Invalid runs	Effective rate	Invalid script	Comm or network error	TCAC S/W or H/W	AFEWES S/W or H/W	ACETEF S/W or H/W	Operator error
DAY 1	31	18%	356 ms	6	81%			1	4		1
DAY 2	37	18%	333 ms	7	81%				3	2	2
DAY 3	46	17%	171 ms	3	93%			1		1	1
DAY 4	26	65%	417 ms	4	85%	1			1		2
DAY 5	19	18%	194 ms	1	95%						1
DAY 6	37	19%	301 ms	2	95%				1		1
DAY 7	35	19%	236 ms	4	89%				1	3	
DAY 8	15	19%	589 ms	2	87%					1	1
DAY 9	9	18%	191 ms	3	67%		1		1		1
Total	**255**			**32**		**1**	**1**	**2**	**11**	**7**	**10**

H/W = hardware ms = millisecond S/W = software

Phase 3 produced 255 total runs over nine days. Of those 255 runs, 32 were aborted yielding an effectiveness rate of 87.5%. The primary causes were federate failures (20) followed by operator procedure problems (10) during test runs. Network problems accounted for one lost run. Some runs were tagged for further analysis since they showed high closed-loop latency (> 500 milliseconds) which exceeded the design limit. These runs were allowed back into the valid test results when the analysts determined that the measures of performance fit within the population as estimated by the good run measures of performance. On-site quality control at AFEWES indicated that all usable engagements provided data consistent with the Phase 1 HITL and Phase 2 tests. In Phase 2, 341 total runs were attempted and 95 runs aborted (a 72.2% effectiveness).

The number of total runs (341) did not include 22 ADS excursion runs and time synchronization runs. Problems encountered with specific components of the federation during Phase 3 testing are explained in the subsections that follow.

4.2.2 Federate Summaries

4.2.2.1 Jammer (ACETEF) Federate

The following problems and anomalies were experienced during the test with the jammer federate.

4.2.2.1.1 ATEWES Problems: Power Level/Waveforms

Problems associated with ATEWES caused numerous aborted runs. During the first day of testing, the pod dropped jamming on an active threat for no apparent reason. The pod also would not reengage threats that had lost track and then reacquired and established good track (<.1 degree error). The first problem appeared to be either a drop in power on the ATEWES or ATEWES was dropping pulses from the waveform causing the pod to misidentify the waveform it received. The second problem appeared to be a logic problem in the HLA gateway or because of ATEWES power or waveform problems. This problem was seen again on three runs during the second day. All runs were completed to diagnose the problem. These runs were all scored as bad. The problem was initially addressed at ACETEF by reconfiguring ATEWES. The ATEWES was comprised of three RF transmitter units. During the FIT on 12 April, the JADS emitters were distributed across all the transmitters. One of the units began producing waveforms with reduced power levels, so ATEWES was reconfigured for all JADS emitters to be on a single transmitter. One of the threat emitters was a continuous wave (CW) radar. This placed a high load on the transmitter. This emitter was moved to the unused (spare) transmitter and this seemed to make the problem manageable. It was seen again later, however the pod continued to respond normally.

A run was declared bad when no jamming output from ACETEF for the SADS VIII was seen at AFEWES. During run 123, ATEWES turned off the jamming response to the SADS VIII. Upon review of the RTI log file and internal ACETEF files (commands sent to ATEWES), no off command was sent. The reason why ATEWES stopped jamming was not determined.

Two runs were aborted during the fourth day of testing when ATEWES appeared to miss a threat mode activation command. One event occurred when the mode activation was 19 milliseconds after a mode off command. It is unknown why ATEWES missed the activation. JADS determined that occasionally the revisit rate of ATEWES was larger than the interval of more than one sequential mode command from a threat. If two modes appeared in the mailbox before ATEWES could check for new messages, it would select the last message and act on it. ACETEF checked the shared memory data exchange for what commands were sent to ATEWES and what ATEWES received. The appropriate commands were sent, but ATEWES did in fact miss an on command that followed receipt of an off command. Multiple instances of erroneous waveforms for the WEST X, the SADS VI, and the SADS VIII were observed at ACETEF. In addition, multiple instances for the SADS III were observed. The SADS III instances were

attributed to residual information in a data structure in the digibus monitor to HLA interface program. Two runs were lost when the DTMS reported an additional SADS III. There was no indication that the emitter was being created in RF and the log files showed that there were no extraneous commands sent to ACETEF by the federation. The digibus monitor was rebooted and the problem did not reappear. The run was repeated successfully. One run was lost when the SADS III signal being sent to the pod was incorrect causing the pod to alternately drop and resume jamming. Tracking error was very low so the pod response should have been continuous. This generally indicated a problem in ATEWES. Testing ended 30 minutes early when it became apparent that the problem was not easily solved. The false SADS VIII was seen frequently. The DTMS rack was located near the receive antenna on the pod in the anechoic chamber. The rack was covered with radar-absorbing material to see if the false SADS VIII was a reflection or due to some component radiating in the rack. That was the only hypothesis on the origin of this signal. The DTMS recorded and passed to the federation the frequency at which the pod identified the threat. A spectrum analyzer monitored the ATEWES output. One run was lost when the pod stopped jamming the SADS VIII. This threat was scripted (unmanned) and tracking error was set to zero. The pod should never stop jamming this threat under these test conditions. This occurred at the same time the pod was reporting an additional SADS VI (discussed below). The combination of the two errors prompted the JADS test controller to abort the run. The pod identified the false SADS VI first at a frequency very near the real SADS VI. However, the analyzer did not indicate that ATEWES produced any such energy. GTRI hypothesized that the pod misidentified the frequency on the initial identification and correctly identified the frequency on all subsequent revisits to that threat. This caused the pod to think that there were two threats instead of one. Since this first SADS VI aged out before the real SADS VI activated, the engagement was not altered, but the correct threat ID MOP was affected. Although there was no hypothesis about its origin, the WEST X problem was sufficiently similar to the SADS VIII that it may have been caused by the same source -- the uncovered rack in the chamber. Personnel continued to look for additional samples of these false readings/emissions. A false WEST X emitter may have affected the two pod internal timing measures since it indicated a different background against which identification must occur. In the reference test condition, the WEST X was not activated. Since this emitter does not register on ADRS, no other MOPs were affected. The frequency identification indicated that it might be an inter-modulation product of two emitters in the scenario. This product was amplified by broadband amplifiers that had to be installed to boost the power level of the WEST X and SADS VIII signals to make the pod respond properly. These emitters provided background noise in the scenario. GTRI recommended reducing the power level slightly on one of the emitters.

When the ATEWES disk capacity filled, one run was aborted. Files were moved to another computer to execute the next run. One run was aborted when a line from ATEWES to the pod became disconnected. This occurred when the F-16 was suspended in the anechoic chamber. One run was lost because the pod receive antennas were disabled. This test was intended to replicate an ISTF test. The F-16 fire control radar (FCR) transmitted to simulate EMI/EMC testing normally seen in an EW test. Disabling the receive antennas violated the reference test condition being represented so the antennas were reconnected and the test proceeded. Two false alarms were noted, which were more than were observed the previous week. However, it was doubtful that these false alarms were due to the FCR. The FCR was not responsible for any false

alarms in the OAR phase of the EW Test. If this test was a real ISTF test, the tester would be required to resolve the source of the false alarms or accept that the false alarm rate was acceptable. The JADS test director determined that the false alarm rate was acceptable for the JADS test.

Latency analysis uncovered one run that had a negative minimum latency. This usually indicated a time synchronization problem. On the day this occurred, closed-loop (threat mode change from AFEWES to ACETEF, jammer response from ACETEF to AFEWES) latency averaged 205 milliseconds. Maximum closed-loop latency was 417 milliseconds.

ACETEF technicians worked late one night to bring ATEWES back on line. JADS and ACETEF made some check runs prior to the scheduled start time on the last day of testing to verify that ATEWES was functional. Formal testing resumed on schedule and a total of four good runs were completed out of the 10 runs attempted. ATEWES failed on the last attempt. ACETEF attempted repairs. The final run was aborted when ATEWES began dropping pulses from the SADS III waveform. The JADS test controller terminated the Phase 3 test an hour later when ATEWES was not restored.

4.2.2.1.2 ATEWES Disk Full

One run aborted when the ATEWES computer disk became full. Files were moved to another computer to execute the run. This was expected to take five minutes. The JADS test controller chose to abort the run and take a 15-minute break instead.

4.2.2.1.3 ATEWES Disconnected

One run aborted when a line from ATEWES to the pod became disconnected. This occurred when the F-16 was being suspended in the anechoic chamber.

4.2.2.1.4 Negative Latency

Latency analysis uncovered one run that had a negative minimum latency. This indicated a time synchronization problem. The clocks were resynchronized and testing continued.

4.2.2.2 Test Control Federate (TCF) and ADRS Operation

The following problems and anomalies occurred during the test with the TCF and the ADRS computers.

4.2.2.2.1 Script Loading Failures and TCF Core Dump

JADS unsuccessfully tried to manually create an aircraft script with a faster air speed. This script resulted in one ADS excursion test run being aborted because of anomalous results reported by AFEWES.

Several runs were delayed when the TCF failed to send the script information from ADRS to the rest of the federates. The cause of this was not determined.

Multiple runs were aborted when TCAC personnel entered incorrect script numbers. This proved to be the most frequent cause of script-related problems. Most of these problems were due to two causes: 1) the ADRS operator entered an invalid profile number, or 2) the profile entered did not have a corresponding script file in the proper directory.

TCF crashed only once during Phase 3 and was the only TCAC federate crash during Phase 3.

4.2.2.2.2 ADRS Crashes

At least one of the ADRS PCs crashed frequently before, during, or after the test run. This usually resulted in leaving an open TCP connection on the TCF federate computer. A TCP connection was a one-to-one communications link defined by the IP addresses of the two hosts and the two TCP port numbers of the sender and receiver. This problem arose from the basic design of how TCP handled intermittent data, lost communications, etc. Typically, the next run was delayed by several minutes while the connection between the two machines "timed-out" and the TCF was restarted. ADRS failed 10 times. Five of these times occurred during 15 consecutive runs. These runs were terminated at the end of the manned threat engagement. The JADS test controller attempted to speed execution by terminating the run after all useful data were collected from the two manned threats, nearly a minute before the end of the platform script. This apparently caused ADRS to become unstable and crash. It should be noted that the remaining five instances occurred when the runs were allowed to continue through the end of the script. This was a higher rate than experienced during the first two days of testing.

4.2.2.3 Platform Federate

The following problems and anomalies were experienced during the test with the platform federate.

4.2.2.3.1 Script Error

One run was aborted when the platform federate failed to load a script. Investigation showed that the specific script required had not been transferred to the computer hard disk along with several others. Another script was selected and execution continued. The missing scripts were transferred to the platform computer between runs.

Script generation errors affected the last salvo from the SADS III. Two runs were deemed invalid because the platform script caused the plane to roll and change direction as the salvo was flying out. This was a violation of the ROE. This same script was run on the SADS VIII/WEST X threat pair, but the event was not noticed. The maneuver occurred late in the engagement after both of those threats are disengaged.

4.2.2.4 Radio Frequency Environment (RFENV) Federate

No problems were experienced during the test with the RFENV federate.

4.2.2.5 Terminal Threat Hand-Off (TTH) Federate

No problems were experienced during the test with the TTH federate.

4.2.2.6 AFEWES Threats Federate

The following problems and anomalies were experienced during the test with the AFEWES federate.

4.1.1.6.1 SADS III

One run was lost when the SADS III engagement was not started on time. The AFEWES test controller missed the cue to begin the engagement. The run was repeated successfully.

4.1.1.6.2 SADS VI

One run was lost when the SADS VI simulator at AFEWES failed and another run was lost when the SADS VI simulator test management center (TMC) at AFEWES failed.

One run was lost when the SADS VI was configured for a dry run when a wet run was being executed. This caused the jamming to be suppressed within AFEWES. The SADS VI was reconfigured and the run was repeated successfully.

4.1.1.6.3 SADS VIII

JADS analysts noted numerous instances where J/S was being reported as a negative number greater than -200. This happened in previous phases where J/S was not supposed to be evaluated (e.g., when the jammer or threat system was off); however, JADS observed several instances when the negative numbers were reported for longer than normal duration and during intervals where the J/S should be evaluated correctly. No runs were aborted. JADS observations were confused because of a slight difference in the engagement sequence caused by less experienced operators. AFEWES confirmed that the system was behaving as it had during the Phase 2 test. The negative J/S was produced by mode changes in the threat. The less experienced operators were making more mode changes. AFEWES offered to suppress the negative J/S, but JADS declined the offer.

One run was aborted when the SADS VIII tracking locked. This was the only instance of the problem. No further action was required.

4.1.1.6.4 WEST X

Early on, the WEST X engagements were considerably different than those seen in the Phase 2 test. Originally this was attributed to a new operator. In particular we noticed that the operators were not using auto mode even on dry runs. The JADS threat site observer discussed this with them. They reported that they were unable to use the auto mode because it was not tracking the aircraft. During the final dry run of the day, the operators used auto mode and JADS observed a series of spikes in tracking error, each growing larger than the previous. AFEWES determined the WEST X was operating correctly. The engagements continued to be different than JADS had seen previously. However, the JADS on-site observer was able to interact with the AFEWES operators and the JADS analysts in the TCAC. The observer described to the operators how the engagement played out in earlier tests. In turn, the operators were eventually able to recreate the same kind of engagement seen in earlier testing. This proved that the earlier results were different because of the difference in operators.

During two runs the WEST X did not report firing events because of a switch failure. This had no effect on any of the MOPs. The switch was replaced.

4.1.1.6.5 Time Synchronization

Negative latency was observed on the TSPI data transmitted from JADS to AFEWES. This was determined to be a time synchronization problem with AFEWES. AFEWES lost time synchronization on one run. Time was reported to be off by .6 seconds. Synchronization was reestablished and the test proceeded. This run was not aborted but it was scored as a bad run and was repeated until a good run completed.

4.1.1.6.6 JETS

One run was lost because the interface between the JETS and the federation dropped off-line. This was repaired for the next run.

One run was lost when the JETS failed to create the correct jamming waveform input to the SADS III. This was corrected and the run was repeated successfully.

4.1.1.6.7 TAMS Software Configuration

On timing synchronization runs, AFEWES had to suppress SADS III output to prevent the pod from responding to the wrong signal. This procedure was implemented on day seven. After the timing run, AFEWES failed to change to the correct software configuration. Two runs were aborted. During the first run, the ATEWES computer needed to be rebooted. The second aborted run was a dry run using AFEWES and JADS.

4.2.2.7 Analysis Federate

No problems were experienced during the test with the analysis federate.

4.2.3 Runtime Infrastructure (RTI)

A SUT jammer technique command (JTC) message was missing in the log file summaries for Phase 3 run 146. JADS analyzed the RTI log files and the *EtherPeek*™ packet sniffer data for this event, and the evidence strongly suggested that the RTI itself was responsible for the loss of an ACETEF JTC message to both AFEWES and JADS. The lost message was the JTC with sequence number 2. The ACETEF log showed this message was time tagged by the jammer federate at 75934819 milliseconds (ms) or 21:05:34.819. However, the logs for the other five federates agreed with the corresponding log file summaries that the message was not received by those federates. The *EtherPeek* packet data collected at ACETEF confirmed the message was transmitted. Packet 3135 contained the copy sent from ACETEF RELDISTR to the RELDISTR on the RFENV host machine at JADS; packet 3136 contained the copy sent to the AFEWES RELDISTR. The *EtherPeek* packet data collected at JADS showed that the copy in outgoing ACETEF packet 3135 was received at JADS as incoming packet 5789. The TCP "acknowledgment" was sent from JADS to ACETEF in outgoing JADS packet 5792. Similarly, the packet data collected at AFEWES showed that the copy in outgoing ACETEF packet 3136 was received at AFEWES as incoming packet 2459, and it was acknowledged by outgoing AFEWES packet 2469. We used the NetSense tool to analyze the TCP traffic for the ACETEF-to-JADS and ACETEF-to-AFEWES TCP connections in the three packet sniffer files. There were no indications of any problems or errors or of excessive traffic at the time that the message was sent. There was evidence that the RTI was responsible for the loss of the message at both JADS and AFEWES.

1) The TCP software in the RELDISTR at both JADS and AFEWES sent a TCP acknowledgment for the message they each received, which implied that the RTI received the content of those messages.
2) The message was not logged by the RTI loggers in the AFEWES, RFENV, platform, TTH, and TCF federates, which implied that the RTI did pass the message content to the federates.

4.2.4 Wide Area Network

The network routers used in Phase 3 had a redesigned Ethernet interface. This contributed to the improved network performance when compared to Phase 2. The following problems and anomalies were experienced during the test with the network.

Several runs from both day 1 and day 2 exhibited a higher than normal latency for a particular message type at the same point in the engagement. The jammer produced two responses in rapid succession that were transmitted to both JADS and AFEWES. These got to the RTI and were logged. These were reliable messages sent TCP/IP. The first message had normal latency. The second message had normal latency to one site but not the other. Network monitoring equipment logs showed that the latency occurred between the time that the RTI was handed the message and

when the message was placed on the network. Either the RTI delayed the message before giving it to the computer hardware or the computer TCP/IP implementation caused the delay. This did not appear to cause an unacceptable latency, however it was monitored throughout the rest of the test.

4.2.5 Test Execution Lessons Learned

During daily testing, a run matrix was used to determine which runs were executed. A set of procedures was used to start the federates and their subcomponents, initiate and stop the run, and shut down the federates. The work done during preliminary testing (e.g., FAT, FIT) provided JADS with a repeatable methodology for orderly federation operation that was used for the formal test. Nonetheless, problems were frequently encountered, errors were made, and unanticipated issues arose. The areas critical to performing distributed testing that yielded valuable lessons learned for testers are discussed below. Lessons learned or solutions are provided based upon JADS test requirements.

4.2.5.1 Software/Hardware Reliability Issues

ADRS equipment crashes and reboots disrupted testing less frequently in Phase 3, which increased the rate of testing. The problem was moderated during Phase 2 by adopting new procedures like rebooting each time a computer was at idle during lunch or outages at another facility. SGI O_2 to PC interface software developed for our federates (called the JADS communicator) would leave a communications socket allocated after ADRS crashed so additional time was wasted waiting for the socket to reset afterwards. Procedural speed for starting ADRS impacted the problem. It was very important that the computer be started in a specific sequence in the TCAC. The action adopted by JADS improved Phase 3 operations. A memory leak problem detected in the ADRS computer was corrected before Phase 3.

4.2.5.2 Test Rehearsal

The FAT and FIT series of federation tests proved invaluable for establishing Phase 3 procedures within the TCAC and with AFEWES and ACETEF. JADS used appropriate test rehearsals and comprehensive integration tests for Phase 3 to become familiar with start-up, execution, and shutdown of federates in a stable, consistent procedure.

4.2.5.3 AFEWES Operator Proficiency and Methodology

JADS TCAC observers recognized significantly different operator performance being used by AFEWES operators in Phase 3 as compared to Phase 2. The SADS VIII/WEST X engagements run during the first week of testing appeared to experience more breaklocks than seen previously. It was determined after discussing this observation with AFEWES that the operators running these systems were different than those used in previous JADS tests and they were unfamiliar with the JADS ROE. Once again, the lesson learned was the criticality of the human-in-the-loop when repeatable test results are needed. JADS did not identify by name which key individuals were

required to perform human-in-the-loop functions and this impacted execution results from each phase of testing.

4.2.5.4 Site Manning/Workload During Test Execution

The number of computers, intricate execution procedures, and high number of test events performed sequentially created a very workload-intensive environment at the TCAC and other locations during testing periods. JADS manning requirements at the TCAC, AFEWES, and ACETEF involved 14 personnel dedicated during the two-week test period. Site manning at AFEWES was increased to four persons (one more than Phase 2) for rotation among stations. JADS reviewed and updated the site manning matrix for Phase 3.

4.2.5.5 Tools and Procedures for Real-Time Analysis of Run "Goodness"

During test runs, the TCAC test controller was highly dependent on ADRS for federation and scenario status monitoring. Analysts were highly dependent on the ADRS emitter state history display for monitoring jammer/threat engagement details. JADS found the analysis federate scenario visualizer to be a solid capability. While it provided an extra set of graphical displays of the unfolding engagement, it also provided real-time feedback of the EW MOPS. Anomalies in miss distance and response times could be instantly assessed, which was an added capability separate from ADRS. The analysis federate could not take the place of an ADRS machine for Phase 3, but it could support troubleshooting of anomalies seen during the runs. Without the analysis federate, problems seen could be at AFEWES, ACETEF, or in one of the many federates run at JADS. The analysis federate aided in the identification of the source of the problem.

If presented with extremely limited time and manning, the analysis federate could be eliminated with only a small impact to test execution. It was not critical to the function of the test, but did provide an extra source of examination of the run execution. The largest benefit of the analysis federate to data analysis was the real-time assessment of the EW MOPS and the integration of the aircraft profile with the threat mode status. If tasks needed to be combined, the analysis federate would need to be updated to assume the responsibilities of the second ADRS machine.

4.2.5.6 Voice Communications

JADS voice links used conference calls with open lines to AFEWES and ACETEF. This was a capability that continued to evolve as command and control requirements evolved. JADS used various head-mounted earphone/microphone equipment and experienced numerous problems in Phase 2 with hearing and being heard across the network. Headset batteries had to be replaced frequently and weak batteries were a major cause of trouble. Problems were alleviated in Phase 3 with equipment improvements and experience. The FAT and FIT demonstrated shortfalls in the voice communications with ACETEF that required more equipment engineering at that facility. Not all personnel at ACETEF had both listening and transmission capability because of the dispersion of personnel into different labs and work areas. Consequently, we learned that message transmission length had to be minimized; external background conversations avoided; and test problem troubleshooting had to be done via a separate line.

4.2.5.7 Network Instrumentation

TSPI data losses between federates occurred less frequently during Phase 3 FIT. The Phase 3 run yield was higher than in Phase 2 when JADS and DMSO investigated data losses. The solution was a fixed join process for federates prior to each test run. After the solution was implemented, data losses in Phase 2 testing were still observed. These losses manifested themselves in the apparent hovering aircraft. Although it was not clear what caused the data loss, dead reckoning aircraft position provided an acceptable solution. However, data loss coupled with the dead reckoning implementation at AFEWES was the suspected cause of an extremely large miss distance value and RTI bundling of federate data for transmission made troubleshooting data flow and transmission problems more difficult. Our tools assessed hardware performance only. A key lesson learned was that instrumentation for federation performance evaluation was inadequate. JADS lacked the ability to examine data passed between local RTI instances. Best effort data could be dropped by the network without notification or without any faults reported by the hardware. Reliable versus unreliable data traffic issues could not be adequately examined.

To improve this in Phase 3, JADS installed nonintrusive network sniffers running data collection and analysis systems at each node. The software was called *EtherPeek* and *EtherHelp*™. Phase 3 network instrumentation was expanded to include network sniffers to monitor network traffic between the sites.

4.2.5.8 Test Control Procedures

JADS researched potential tools for improving situational awareness for network health and readiness across sites and formalized voice protocols and procedures.

4.2.5.9 Software Changes

A big lesson learned was that configuration changes on tools (analysis federate, ADRS display, ACETEF joining process, AFEWES dead reckoning algorithms) might have severe impacts. Configuration changes, even seemingly trivial ones, must be coordinated at all levels. JADS documented formal configuration management procedures for Phase 3 and enforced their use.

4.2.5.10 Latency and Time Synchronization

JADS was highly dependent upon time synchronization of all federate computers and software. However, any requirement for synchronization required the ability to verify the requirement was being met. For example, if one millisecond synchronization accuracy was required, then a capability to measure time between two computers at one millisecond precision was necessary. However, software tools were not available to measure accuracy at that level nor to measure latency and time synchronization in real time. In fact, testing time synchronization across the federation was more art than science. Even with time synchronization and the time cards implemented in all computers, we still found instances where time synchronization slipped, affecting latency measurements. A few occurrences of time synchronization problems across the

federation were observed requiring JADS to research particular runs after daily testing. JADS consistently followed documented time synchronization procedures and hardware settings. Site support personnel were relied upon to implement procedures and verify settings daily.

4.2.5.11 Run Speed/Time Between Runs

Operator boredom due to repetitiveness of test runs at each site may have contributed to run differences. JADS attempted to minimize turn-around time between runs to execute runs as quickly as possible. The time between runs from start time to start time was usually 6 to 8 minutes.

4.2.5.12 RTI Heartbeat

This health check was inadequate for monitoring JADS federation status for several reasons. First, its time scale, which was about 20 seconds before any indication of a problem, was too long for a real-time federation. Second, because it sent its messages via TCP/IP, this system could not detect a problem for federate messages sent via the RTI best effort, i.e., user datagram protocol (UDP)/internet protocol (IP), unless the underlying cause of the problem affected both of those protocols. And third, the RTI did not time stamp and log these messages, so they were perishable and only available in real time. JADS did not use this as a primary indication of federation health so no changes were required. Future federations should investigate RTI tuning features (e.g., RID file parameters) or other RTI management features (e.g., management object model calls) if the federation doesn't implement its own health monitors like JADS.

4.2.5.13 Federate Link Health Check (LHC)

JADS link health scheme provided reasonable insight into federation health once the content and meaning of the message was understood. Analysis has shown that there was a high correlation between the loss of LHC messages and most, but not all, events that involved the loss of other federate messages sent best effort and/or the delay of messages sent via the RTI reliable TCP/IP-based communications protocol. Due to its 1 Hz message frequency, the LHC system sometimes missed best effort data loss events lasting less than 1 second, but those events apparently did not cause any simulation problems. Since the LHC system sent its messages via the best effort protocol, it also did not detect short-duration problems that affected only the TCP/IP connection used for reliable protocol between two federates.

Perhaps, the most interesting result from the post-test analysis of the LHC messages was that the LHC system detected selective, one-way, best effort data losses between federates that may be symptoms of some problem(s) with the RTI using IP multicast groups. For the runs during which these problems were observed, the losses were selective because LHC messages (and usually other federate messages sent best effort as well) were lost between one or more federates at JADS and the federate at another test node but not between the remaining JADS federates and that remote federate. The losses were one-way because the LHC messages between the federates experiencing the problem were lost in only one direction. Typically, during such events, there was no delay in the flow of reliable messages between those federates, if any reliable traffic was

present. It is difficult to understand how network or network hardware problems could produce such selective, one-way data losses. While LHC as implemented had limitations, it was sufficient for JADS in Phase 3. Future federations should consider the limitations noted above if they choose to pursue a similar health monitor scheme for their federation.

5.0 Data Analysis

The test collected two classes of performance data: jammer MOP and ADS measures. The detailed jammer MOP results are covered in a separate classified report that contains both the Phase 2 and Phase 3 data analysis. Results of jammer MOP correlation across EW Test phases are presented in Section 6. Since the JADS test program was designed to evaluate the utility of ADS for EW T&E, the analysis of jammer performance was not the primary focus of the test. Instead, the study of the impacts of ADS on the jammer MOPs provided a method to assess the utility of ADS within the JADS EW Test environment. This section addresses the detailed analysis of the ADS measures.

5.1 ADS Measures

This section describes the relationship between the EW Test objectives for Phase 3 and the ADS measures that ultimately support established JADS-level objectives and issues. Table 8 shows how it was anticipated that EW Test activities would provide information for addressing JADS objectives from an EW perspective by showing corresponding EW and JADS objectives. Table 9 lists the individual JADS-level ADS measures evaluated during Phase 3. The JADS EW Test Phase 3 (ISTF) ADS measure results are compared to those obtained from Phase 1 and Phase 2 testing where appropriate.

Table 8. JADS and EW SPJ Test Objectives Correspondence Matrix

SPJ Obj #	Self-Protection Jammer (SPJ) Test Objectives	Expected JADS Objectives Supported by SPJ Test (May 96 EW Test APA)	JADS-Level ADS Measures Supported*
1	**Measure SUT performance data in each environment**	**Subobj 1-2-2**: Assess ADS capability to support live T&E planning and test rehearsal	
1-1	OAR (Baseline)		
1-2	DSM		1-2-2-2, 1-2-2-3, 1-2-2-4
1-3	ISTF		
2	**Establish repeatability of OAR and ADS test results**	**Subobj 1-2-2**: Assess ADS capability to support live T&E planning and test rehearsal	
2-1	OAR (Baseline)		
2-2	DSM		1-2-2-2, 1-2-2-3, 1-2-2-4
2-3	ISTF		
3	**Correlate data between environments**	**Subobj 1-2-2**: Assess ADS capability to support live T&E planning and test rehearsal	
3-1	OAR-DSM duplicated threats		1-2-2-2, 1-2-2-3, 1-2-2-4
3-2	OAR- ISTF duplicated threats		1-2-2-2, 1-2-2-3, 1-2-2-4
3-3	OAR-HITL duplicated threats		
4	**Quantify the effects of ADS-induced errors**	**Obj 1-1**: Assess validity of data from tests utilizing ADS **Subobj 1-2-2**: Assess ADS capability to support live T&E planning and test rehearsal **Subobj 2-1-2**: Assess network and communication performance constraints and concerns	
4-1	Latency on ADS test results		2-1-2-4
4-2	Effects on human perception		1-1-0-3, 1-1-0-4
4-3	Others		
5	**Measure ADS network performance**	**Subobj 2-1-2**: Assess network and communication performance constraints and concerns **Subobj 2-1-3**: Assess the impact of ADS reliability, availability, and maintainability	2-1-2-1, 2-1-2-2, 2-1-2-3, 2-1-3-3
6	**Measure ADS reliability**	**Subobj 2-1-2**: Assess network and communication performance constraints and concerns **Subobj 2-1-3**: Assess the impact of ADS reliability, availability, and maintainability	2-1-2-2, 2-1-2-3, 2-1-3-1, 2-1-3-2, 2-1-3-3

* JADS-level ADS measures not listed were assessed indirectly after test completion

Table 9. JADS Measures Evaluated During Phase 3

JADS EW Test ADS Measure	Title
1-1-0-3	Degree to which test participants were able to distinguish between ADS (virtual or constructive) versus live assets
1-1-0-4	Degree to which test actions were impacted because of the ability to distinguish between ADS and live (non-ADS) targets
1-2-2-2	Degree to which test control procedures and tools are impacted by ADS and how ADS can impact the pretest development and rehearsal of test control procedures
1-2-2-3	Degree to which data management procedures and tools are impacted by ADS and how ADS can impact the pretest development and rehearsal of data management procedures and tools
1-2-2-4	Degree to which data reduction and analysis procedures and tools are impacted by ADS and how ADS can impact the pretest development and rehearsal of data reduction and analysis procedures and tools
1-2-3-3	Degree to which ADS can increase test times, events, etc.
2-1-1-1	Degree to which live, virtual, and constructive entities exist, can be instrumented, and can be readied for a test
2-1-2-1	Average and peak throughput available for each link
2-1-2-2	Percentage of complex data types received out of order by a federate
2-1-2-3	Percentage of total complex data types subscribed to by a federate that were received by the federates
2-1-2-4	Average and peak data latency
2-1-3-1	Degree to which test events (trials) were affected by ADS components (failure or otherwise) exclusive of network problems
2-1-3-2	Degree to which test events (trials) were affected by network problems (failure or otherwise)
2-1-3-3	Degree to which test events (trials) were affected by personnel problems
2-2-1-4	Ease with which data can be retrieved, post-trial, from a given node
2-2-2-1	Degree to which test managers can control the configurations of ADS participants, the ADS environment data, and ADS networks
2-3-2-3	Degree to which protocols, processes, and procedures are needed to enable effective centralized test control
2-3-2-4	Degree to which real-time analysis systems support test safety and other test control requirements

Detailed descriptions of these JADS-level ADS measures and Phase 3 test results are provided in the subsequent paragraphs.

5.1.1 Measure 1-1-0-3. Degree to which test participants were able to distinguish between ADS (virtual or constructive) versus live (non-ADS) assets.

Intent. The intent of this measure was to determine if workstation operators at AFEWES could distinguish between ADS-linked assets and non-ADS assets (SPJ installed at AFEWES) and, if so, to what extent.

Data Collection and Analysis Approach. Interviews were conducted with AFEWES test participants concerning their perceptions and actions. Interview questions focused on procedural and technical differences between the EW ADS test and other non-ADS test events with which participants had experience. AFEWES operators were asked to describe the impacts of any unusual procedures or unrealistic behaviors on their ability to perform test operations. JADS analysts reviewed and summarized recorded remarks, also noting differences between Phase 2 and Phase 3 where appropriate.

Data Sources. Nine interviews were conducted. Interviewees included the AFEWES federate controller, the TAMS operator, the JETS operator, and the SADS III, SADS VI, SADS VIII, and WEST X threat system operators. All Phase 3 interviewees had gained experience with ADS testing during Phase 2.

Results. The ADS testing experience gained from Phase 2 participation may have made the AFEWES operators feel comfortable with ADS test processes and procedures. Few of them noted any procedural differences between Phase 3 ADS and traditional testing methods; while for Phase 2, several had noted the increased need for strict equipment configuration control and checklist adherence while setting up for ADS runs. On the other hand, most of the Phase 2 technical differences reappeared, although not as frequently. AFEWES operators noted far fewer system crashes, link problems, and federation interface software problems than occurred during Phase 2 testing. There was still some unusual target behavior identified for all four threat systems, typically, the target ceasing to move or jumping out of gates, which made it difficult for the threat operators to track. This unusual behavior occurred predominantly on high-speed excursion runs or on runs that were aborted because of malfunctioning equipment, network link outages, or procedural mistakes. The TAMS operator called attention to the fact that the scripted ADS target data used for Phase 2 and Phase 3, with only two basic target scenarios, were more repetitive than that typically used.

Conclusions/Recommendations. While AFEWES operators noted slight procedural and technical differences that enabled them to distinguish ADS testing from non-ADS testing, there did not appear to be any major issues or problems stemming from the differences. Most of the system crashes, link problems, and federation interface software problems that provided distractions and caused run losses during Phase 2 testing were alleviated with the use of a new RTI version in Phase 3.

5.1.2 Measure 1-1-0-4. Degree to which test actions were impacted because of the ability to distinguish between ADS and live (non-ADS) assets.

Intent. The intent of this measure was to determine if being able to distinguish between ADS-linked assets and non-ADS assets impacted AFEWES workstation operators' actions, particularly actions that affected SUT MOPs.

Data Collection and Analysis Approach. Interviews were conducted with AFEWES test participants concerning their perceptions and actions. Interview questions focused on procedural and technical differences between the EW ADS test and other non-ADS test events with which participants had experience. AFEWES operators were asked to describe the impacts of any unusual procedures or unrealistic behaviors on their ability to perform test operations. JADS analysts reviewed and summarized recorded remarks, also noting differences between Phase 2 and Phase 3 where appropriate.

Data Sources. Nine interviews were conducted. Interviewees included the AFEWES federate controller, the TAMS operator, the JETS operator, and the SADS III, SADS VI, SADS VIII, and WEST X threat system operators. All Phase 3 interviewees had gained experience with ADS testing during Phase 2.

Results. Here again, Phase 2 and Phase 3 interview responses were similar. AFEWES operators indicated no negative impacts of ADS test procedural differences in their ability to perform test operations, but a few minor performance impacts because of ADS technical differences were cited. As in Phase 2, both the SADS VIII and WEST X operators had difficulty tracking high-speed targets when they 'jumped out of gates" during excursion runs, and all threat system operators noted occasional erratic target behavior prior to runs being aborted, which made tracking difficult. However, these behaviors were experienced on far fewer runs during Phase 3 than during Phase 2.

Conclusions/Recommendations. Some AFEWES threat system operators noted technical differences between ADS testing and non-ADS testing that impacted their ability to track targets and collect SUT data. Phase 3 SUT MOP analysis was performed with this knowledge, as it was for Phase 2. JADS analysts noted all AFEWES operator log comments and carefully researched any potential SUT data anomalies.

5.1.3 Measure 1-2-2-2. Degree to which test control procedures and tools are impacted by ADS and how ADS can impact the pretest development and rehearsal of test control procedures.

Intent. This measure was intended to evaluate the impact of ADS on test control procedures including development and rehearsal by comparing test control procedures for ADS versus non-ADS testing.

Data Collection and Analysis Approach. Interviews were conducted with Phase 3 test controllers and test executors to ascertain their perceptions about ADS test control procedures. Interview questions required an assessment of the quality and complexity of ADS test control procedures, as well as the potential differences between ADS versus non-ADS test control

procedure development and rehearsal. JADS analysts reviewed and summarized recorded remarks, also noting differences between Phase 2 and Phase 3 where appropriate.

Data Sources. Seven interviews were conducted. Interviewees included the JADS test controller positioned in the TCAC, three JADS test executors positioned at AFEWES and ACETEF, and three test station operators positioned in the TCAC.

Results. Interview responses for Phase 3 did not differ much from Phase 2 responses, except to note some improvement in Phase 3 test control because of stricter adherence to procedures and improved voice communications equipment and protocols. These changes were implemented in response to numerous run losses during Phase 2 due to miscommunication. Test control procedures included those used for pretest coordination, voice communications initialization, time synchronization and network verification tests, federation joining, and test event start-up procedures. Again, procedure rehearsal was conducted prior to test execution during integration and acceptance testing events. Most interviewees noted that experience with Phase 2 execution benefited Phase 3. Checklist procedures were adapted to provide tighter test control and better communication among sites based on lessons learned during Phase 2. Some new procedures were also developed to enable the use of *EtherPeek*, a new data collection tool employed during Phase 3.

Conclusions/Recommendations. For the most part, because of procedural improvements, test control in Phase 3 did not have to be as flexible as it did during Phase 2. Fewer procedures had to be developed informally during Phase 3 execution to alleviate miscommunication or enhance the federation execution process, since improvements, such as better voice communications equipment, were made in response to Phase 2 issues. The general consensus among respondents from their experiences with ADS testing was that test control procedure development and rehearsal do not differ much from that required for non-ADS testing, but were dependent primarily on the requirements of the particular test, whatever the type.

5.1.4 Measure 1-2-2-3. Degree to which data management procedures and tools are impacted by ADS and how ADS can impact the pretest development and rehearsal of data management procedures and tools.

Intent. This measure was intended to evaluate the impact of ADS on data management procedures and tools including development and rehearsal by comparing data management procedures and tools for ADS versus non-ADS testing.

Data Collection and Analysis Approach. Interviews were conducted with Phase 3 data managers and analysts to ascertain their perceptions about ADS data management procedures and tools. Interview questions required an assessment of the quality and complexity of ADS data management procedures and tools, as well as the potential differences between ADS versus non-ADS data management procedures and tools development and rehearsal. JADS analysts reviewed and summarized recorded remarks, also noting differences between Phase 2 and Phase 3 where appropriate.

Data Sources. Seven interviews were conducted. Interviewees included the TCAC data manager, the AFEWES and ACETEF data management representatives, and four JADS data analysts.

Results. Phase 3 interviewees noted that the same data management procedures and tools developed and implemented for Phase 2 testing were reused for Phase 3 test data management. These included the EW federate data loggers, visualization tool data archiving software, and various UNIX and PC-based file transfer protocols. The main difference between the two phases was better documentation of data management procedures for Phase 3, including a formal written data backup plan. Essentially, Phase 2 experience consolidating, transferring, and storing data served as rehearsal for the analysts who had an even greater number of data types to track and store during Phase 3. Again, respondents surmised that data management efforts would differ significantly between ADS and non-ADS testing. The detailed coordination required across distributed sites and the electronic transfer, storage, and accurate retrieval of multiple different types of data from distributed federates were cited as major distinctions.

Conclusions/Recommendations. The conclusions and recommendations derived from Phase 3 interviews echo those from Phase 2. Regardless of the type of testing conducted, i.e., ADS versus non-ADS, the key to data management is developing and rehearsing a plan that sufficiently addresses the consolidation, transfer, storage, and retrieval of the data needed for sound SUT analysis. Tools can be developed or acquired to meet particular data management needs, whether for small or large amounts of data, and whether generated by a single or multiple facilities. It is careful planning and rehearsal that ensure data management processes run smoothly when limited test time is available and personnel are busy with other issues.

5.1.5 Measure 1-2-2-4. Degree to which data reduction and analysis procedures and tools are impacted by ADS and how ADS can impact the pretest development and rehearsal of data reduction and analysis procedures and tools.

Intent. This measure was intended to evaluate the impact of ADS on data reduction and analysis procedures and tools including development and rehearsal by comparing data reduction and analysis procedures and tools for ADS versus non-ADS testing.

Data Collection and Analysis Approach. Interviews were conducted with Phase 3 data analysts to ascertain their perceptions about ADS data reduction and analysis procedures and tools. Interview questions required an assessment of the quality and complexity of ADS data reduction and analysis procedures and tools, as well as the potential differences between ADS versus non-ADS data reduction and analysis procedures and tools development and rehearsal. JADS analysts reviewed and summarized recorded remarks, also noting differences between Phase 2 and Phase3 where appropriate.

Data Sources. Seven interviews were conducted. Interviewees included the TCAC data manager, the AFEWES and ACETEF data management representatives, and four JADS data analysts.

Results. The data reduction and analysis procedures developed and implemented for Phase 2 testing were reused during Phase 3 for reducing and analyzing test data, with the addition of several commercial analysis packages. Reused tools included the EW federate data loggers, visualization tools and SUT data reduction software (e.g., ADRS, analysis federate), various C++ log file summary and comparison software utilities, and various PC-based statistics and analysis packages (e.g., Excel®). Additional Phase 3 tools included *EtherHelp* data packet analysis software and Analytical Software Corporation's *Statistix* application. Reduction and analysis of Phase 2 data served as a rehearsal for the Phase 3 effort; in fact, spreadsheet templates created for Phase 2 data reduction were employed to reduce the amount of work for Phase 3. In addition, integration and acceptance testing events gave analysts an opportunity to work with new Phase 3 tools and data formats. Interview respondents reiterated that with ADS testing, it might not always be possible to rehearse automated reduction system use without participation from personnel at all involved sites. On the other hand, ADS testing does provide a benefit over non-ADS testing in that electronic data can be obtained for reduction and analysis from distributed sites within moments after test completion.

Conclusions/Recommendations. The conclusions and recommendations derived from Phase 3 interviews echo those from Phase 2. Regardless of the type of testing conducted, i.e., ADS versus non-ADS, the key to data reduction and analysis is developing and rehearsing a plan that sufficiently addresses the data analysis objectives of that test. Tools can be developed or acquired to meet particular data reduction and analysis needs, whether for small or large amounts of data, and whether generated by a single or multiple facilities. Rehearsal can help analysts become familiar and comfortable with the tools they will be using and ensure that the tools are capable of handling the type and amount of raw data collected, so that the data can be successfully manipulated to provide meaningful SUT results.

5.1.6 Measure 1-2-3-3. Degree to which ADS can increase test times, events, etc.

Intent. This measure was designed to determine how ADS could increase test time and events. Comparison of the time required to collect a number of test events (trials) was made between ADS and non-ADS phases of the EW Test.

Data Collection and Analysis Approach. JADS analysts reviewed detailed information contained in written test control, event, and problem logs to determine the number of trial events completed during Phase 3, as well as the amount of time spent actively testing and how that time was spent.

Data Sources. Data sources for this measure included the detailed test controller log, test event log, and the hardware, software and network problem log (HSNPL). Information recorded in these written test logs included daily start and stop times, personnel break start and stop times, run start and stop times, run outcomes, problem start and stop times, and detailed notes on the impacts of any problems experienced.

Results. The total active test time during Phase 3 was 48.8 hours out of a total scheduled test time (including breaks) of 62.6 hours over nine days. A total of 270 trial runs were completed,

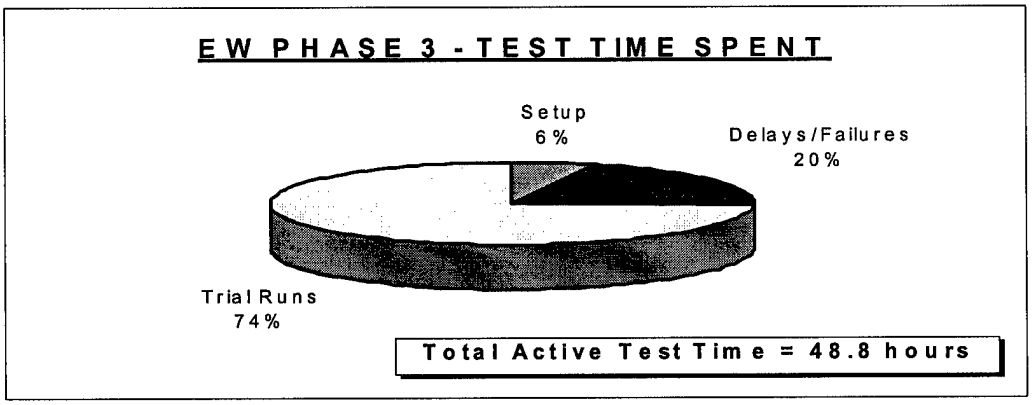
including 15 excursion or special test runs and 32 aborted runs. The remaining 223 trials were considered successful test events for providing valid SUT data, resulting in an average trial success rate of 87%. Table 10 presents a summary of the trials completed during Phase 3 testing and the daily test time required.

Table 10. EW Test Phase 3 Test Time and Run Summary

	Total Test Time	Active Test Time	Total Trial Events	Number Special Trials	Number Aborted Trials	Number Successful Trials	Trial Success Rate*
13 April	7:57:00	6:21:00	31	0	6	25	81%
14 April	7:08:00	5:19:00	37	0	7	30	81%
15 April	7:59:00	6:21:00	46	0	3	43	93%
16 April	7:27:00	5:44:00	35	9	4	22	85%
19 April	7:38:00	5:53:00	21	2	1	18	95%
20 April	7:50:00	5:52:00	37	0	2	35	95%
21 April	7:36:00	5:53:00	35	0	4	31	89%
22 April	6:40:00	5:05:00	18	3	2	13	87%
23 April	2:20:00	2:20:00	10	1	3	6	67%
TOTALS	**62:35:00**	**48:48:00**	**270**	**15**	**32**	**223**	**87%**

* Trial success rate does not incorporate special trials

Figure 12 shows the percentage of active test time allocated to the following categories during Phase 3: test setup, delays/failures, and trial run performance.



*Test setup activities included time synchronization checks and pretest communications check-out procedures

Figure 12. EW Test Time

Table 11 shows a comparison of scheduled test time and number of test events completed between Phase 3 and previous test phases, including OAR and HITL.

Table 11. EW Test Events by Phase

	Schedule Test Hours***	Actual Test Hours***	Total Trial Events***	Number of Completed Trials	Number of Useable Trials
OAR	18.4 hours	14.4 hours*	136**	136	126
HITL	64.0 hours	56.0 hours	341	267	199
Phase 2 (ADS)	71.0 hours	57.2 hours	363	246	245
Phase 3 (ADS)	68.2 hours	48.8 hours	270	223	223

* Includes 3.5 hours of test time from the final two OAR risk reduction events
** Each OAR trial provided SUT data for four active threats; data essentially equivalent to that produced by two trial events during other phases
*** Test hour and trial event totals within ADS phases included excursion runs

Of the 270 trial events completed during Phase 3 testing, 223 were considered to provide usable data for SUT evaluation. The total scheduled test time required to obtain these data was just over 62 hours across 9 days; however, only 48.8 hours were spent actively testing. Of the time spent actively testing, 20% (10 hours) was lost because of delays and system failures.

Conclusions/Recommendations. Even though it appears the OAR test events were more successful in number of useable trials per scheduled test hour, these data may be misleading because of the time required to execute each scheduled test hour on an OAR. The HITL and ADS test phases appear to consume more time per usable trial, but these data were collected over a significantly shorter test period. The OAR test took four calendar months to collect the 14.4 hours of data because of the nonavailability of OAR test time. It was not possible to conduct 14.4 consecutive hours of test time on the OAR. In the HITL and ADS test phases, the data were collected over a much shorter time period, but the impact of problems and anomalies was much more severe.

5.1.7 Measure 2-1-1-1 Degree to which live, virtual, and constructive entities exist, can be instrumented, and can be readied for a test.

Intent. This measure provided an assessment of the availability of the required ADS components to support the infrastructure (equipment, personnel, technical experts, cost, etc.) for the EW SPJ Test.

Data Collection Plan and Analysis Approach. JADS performed an assessment of the RTI interface logger and the requirements needed to transfer data to the logger.

Results. To accurately calculate latency of all messages in the JADS EW Test federation, the development team determined that data needed to be recorded at each federate. For this reason, the RTI interface logger was developed. The RTI interface logger resided between a federate and its local RTI component. It recorded all data passed to/from the RTI. Each attribute and

interaction was time stamped as it went into or came out of the RTI. The logger was linked into each federate in the JADS EW Test federation.

The logger software was relatively easy to develop. For every function defined in the RTI interface, there was a corresponding function in the logger. It took about two staff-months to develop and test the logger software. When the RTI interface specification changed, a new version of the logger had to be developed. Once the software was developed, configuring a federate to use the software was trivial. Depending on the federate design, about 10 to 20 lines of code must be modified to convert a federate to use the logger. The logger can be used with any federate as long as the RTI and logger versions match (e.g., if RTI 1.3 is used by the federate, then V1.3 of the logger must also be used).

Since the file created by the logger was in binary format, the major complexity in data analysis was in writing the software that read the log file. Although the logger can be used with any federate, special log file reader software must be written to translate the binary data to human-readable format. JADS software analysts developed log file reader software to

- Produce log file summary statistics (data counts and latency by attribute/interaction)
- Create American Standard Code for Information Interchange (ASCII) data files used by other analysis software (e.g., ADRS)
- Create a list of threat mode changes and jammer responses
- Calculate attribute and interaction publish and receive data rates
- Display attribute and interaction header and log time values

An important part of each data type was the header. The header included a field containing the time that the RTI created the message. Latency from data creation to consumption was easily determined by calculating the difference between the header time from the sending and the log time from the receiving loggers. Another important element of the header was the sequence number. A separate sequence number was maintained for each data type within each federate. It was easy to determine data dropouts by noting missing sequence numbers for a particular attribute or interaction.

Other instrumentation needed for the AFEWES threats was provided by the facility. No unique instrumentation was needed for threats. Measurements of internal facility latency were conducted post-test at AFEWES.

Comparatively, the instrumentation available for the OAR and HITL tests needed fewer modifications than the new developments required to execute the ADS phases. Instrumentation was not available to collect data for all ten MOPs on the OAR, which forced the need for the execution of the HITL and SIL tests. In all test phases, instrumentation was not completely ready to fully support the EW Test.

Conclusions/Recommendations. Ease of instrumentation of ADS components is dependent on early planning and design for flexibility. More than a year before the simulation (federate) software was developed, the SPJ team was thinking about data collection and analysis. This

planning influenced the design of the federates. Specifically, the time and sequence number were added to the data headers in the ICD to simplify the calculation of latency and determine dropouts.

Data collection requirements changed as the test progressed. The logger was developed to provide flexibility in data collection. It could be configured to log all interface events with the RTI, only certain types of events, or it could be turned off entirely. When it was linked with a federate it required very few modifications to the federate software. It logged the data without impacting the execution of the federate. The logger could be linked with all federates in a federation, collecting data at each node. It could be selectively linked with specific federates or set up as a stand-alone logger federate.

The logger was developed to collect data based on the interface to the RTI without regard for the type of data published by the federates that used the logger. This allowed the logger to be developed independently of the federate software. Changes to the logger had no impact on the federate software. The current version of the logger takes advantage of specific features of SGI computers. The logger software could easily be modified to remove the SGI features without any modification of federate software.

Instrumentation to measure internal facility latency was necessary for understanding the complete latency picture. However, traditional testing within the facility may not require this type of instrumentation, so it may not be available to ADS-based tests.

5.1.8 Measure 2-1-2-1. Average and peak throughput available for each link (JADS to AFEWES, JADS to ACETEF, and AFEWES to ACETEF).

Intent. The intent of this measure was to provide an indication of the amount of data traffic that was sent across each link for Phase 3 testing, as well as the amount of available bandwidth utilized to send this level of data traffic.

Data Collection and Analysis Approach. Data traffic over each network link was actively monitored during Phase 3 testing using *SPECTRUM.* Analysts used *SPECTRUM* to model the Phase 3 network and query network equipment for traffic and performance information at 30-second intervals. JADS analysts reviewed collected data for statistical trends and anomalies. Anomalies were further tracked and researched to determine any impact on collected SUT data.

Data Sources. The *SPECTRUM* tool provided a near real-time capability for network traffic monitoring, presenting current packet rate and load (% bandwidth utilized) information, as well as packet error and discard rate information, for Phase 3 network equipment. *SPECTRUM* recorded captured query information to database for later analysis.

Results. The average and peak packet rate and load values experienced for each Phase 3 network link are presented in Table 12. These values encompass 48.8 hours of active testing over nine days. The peak packet rate and utilized bandwidth values were all captured on the fourth day of test execution while excursions were run with four active threats.

Table 12. EW Test Phase 3 Network Link Performance

	Packet Rate		Bandwidth Utilized	
Network Link	**Average**	**Peak**	**Average**	**Peak**
JADS-AFEWES	45.21 /sec	315 /sec	5.53%	65%
JADS-ACETEF	26.74 /sec	85 /sec	2.73%	10%
AFEWES-ACETEF	20.67 /sec	151 /sec	2.51%	21%

Conclusions/Recommendations. Average and peak packet rate and load values for Phase 3 execution were very similar to those from Phase 2 and fell within expected levels for each network link. The highest levels of traffic were observed between JADS and AFEWES, corresponding to the largest amount of information that had to be shared between these two sites. Average load values show that <6% of the available bandwidth on each T-1 line was typically used to pass data between distributed sites. The maximum bandwidth utilized for any link rose to 65% of total capacity, but this peak occurred while four threat systems were active simultaneously during excursion runs.

5.1.9 Measure 2-1-2-2. Percentage of complex data types received out of order by a federate.

Intent. This measure was intended to determine the percentage of complex data types received in a different order than originally sent out by a federate.

Data Collection and Analysis Approach. The RTI interface log files were used to record all published and subscribed complex data types for each federate. JADS analysts reviewed, summarized, and compared all collected log file data. The order of all complex data types received at a federate was compared to the order in which those subscribed complex data types were published by the sending federate to determine if the complex data types were received in the same sequence or order in which they were sent. One additional tool, the *EtherPeek* data packet sniffer and analysis package, was added to the data collection and analysis tool set for Phase 3.

Data Sources. The RTI interface loggers collected published and subscribed complex data types for each federate. Analysis of *EtherPeek* sniffer data post-test shed insight into the behavior of RTI software, federate software, and network in transporting data packets and helped analysts determine the cause for any out-of-order data.

Results. After in-depth analysis of out-of-order data in Phase 2, using the available log file summary and comparison tools, Phase 3 results were not unexpected. Again, no traditional examples of out-of-order data packets were discovered for Phase 3 runs. In essence, that is, for any individual complex data type, no packets were logged leaving a federate in one order and arriving at another federate in a different order. However, there were again instances of out-of-order packets according to a broader definition of out-of-order. In particular, three different types of out-of-order packets occurred.

1) Out-of-Order Data Within a Federate. Complex data type messages generated by federate software incorporate header times to represent their creation time. Each message was also logged and time stamped by the federate logger as it left the federate. Comparison of packet header times to logger time stamps for several messages generated consecutively showed multiple examples where log file time stamp order did not match the order in which the packets were originally generated. The most likely cause of this out-of-order data within an individual federate has been attributed to the multithreaded nature of the federate software processes themselves.

2) Differential Delay of Complex Data Types to Receiving Federates. Another instance of unusual data packet ordering behavior was discovered when analysts compared the arrival times at different federates of a complex data message type generated at one federate. In many instances, the data message arrived and was logged at one site several seconds before being logged at another site. There were two general causes for this behavior. In reliable transmissions the message is passed in serial fashion from the publisher to each subscriber. In best effort this phenomenon is due to differences in network latency between the paths. Although intuitively, this packet behavior could cause simulation visualization or other SUT performance measure anomalies, no negative impact on performance measures or real-time visualization occurred for Phase 2 or Phase 3 because the primary visualization and data collection tool utilized for EW testing (ADRS) based all relevant data presentation and collection on message header time not message receipt time.

3) Associated Complex Data Types Sent Via Different Transport Protocols. Yet another example of oddly ordered data traffic relates to the out-of-order arrival of different, yet associated, complex data types sent from a single federate to another federate. According to the EW Test ICD, each particular complex data message type traverses the network via a designated network transport protocol, typically TCP (reliable) or UDP (best effort). If associated messages, of different complex data type, are generated by a sending federate in a certain order, it is possible for them to arrive out-of-order at the receiving end because of the differential speeds of the associated transport protocol. Odd log file discrepancies were detected as a result of the RTI handling such out-of-order data. Analysts deem that making use of the RTI time management functions might alleviate this out-of-order behavior.

Conclusions/Recommendations. In the strictest sense, the EW Test network and RTI did not cause any out-of-order packet data for Phase 3. However, in the broader sense, out-of-order packet issues discovered through summary and comparison of the log file data collected by each federate during Phase 2 were again detected in Phase 3 analysis. Some of these issues impacted packet traffic order as it was sent, others impacted the order received. Regardless, analysis showed that individual federate software code, the transport protocols utilized by different complex data types, or even the particular RTI functionality selected for use in federation communication can have grave impacts on the order in which data are sent and received by federates. Out-of-order data, in turn, could cause serious SUT performance data anomalies if data collection, analysis, and real-time display tools are not developed with the potential for this out-of-order data in mind. Experimentation with the RTI time management functions might shed some insight into potential ways of alleviating anomalous out-of-order packet behaviors, although

this RTI functionality was not implemented during Phase 2 or Phase 3 for fear of the unacceptably high data latencies that would likely result. Increased data packet instrumentation at each federate via the use of an *EtherPeek* data packet sniffer did provide helpful detail for analysis into the cause of out-of-order data events.

5.1.10 Measure 2-1-2-3. Percent of total complex data types subscribed to by a federate that were received by the federates.

Intent. This measure was intended to report the percentage of complex data types that were lost while traversing the ADS network.

Data Collection and Analysis Approach. The RTI interface log files were used to record all published and subscribed complex data types for each federate. JADS analysts utilized software tools to summarize and compare log file contents and identify lost complex data types between publishing and subscribing federates. One additional tool, the *EtherPeek* data packet sniffer and analysis package, was added to the data collection and analysis tool set for Phase 3.

Data Sources. The RTI interface loggers collected published and subscribed complex data types for each federate. Real-time network instrumentation files and test observer notes were additional sources of information as to the potential cause of certain data losses. Analysis of *EtherPeek* data post-test shed insight into the behavior of RTI software, federate software, and network in transporting data packets and helped analysts find the cause of data losses as well as determine the location where the data were lost.

Results. As in Phase 2, the analysis of lost data was approached from both a perspective of number of messages lost and loss duration. For the first approach, lost messages were tallied for six complex data message types across relevant network links. The six types were selected for evaluation based on their ability to provide insight into the impact of lost traffic on SUT data validity. In other words, these were the message types that, if lost, should have had the most noticeable effect on SUT behavior and the SUT performance measure data collected. There was a single reliable message loss event, from ACETEF to both JADS and AFEWES; whereas no reliable traffic was lost during Phase 2. Phase 3 performance was better than Phase 2 for best effort data traffic because far fewer messages were lost. Only two individual runs experienced any data losses of these message types for Phase 3 compared to thirty-eight runs with unusual data losses (> 5 messages lost) during Phase 2. Table 13 provides a summary of this lost data traffic categorized by message type and network link.

Table 13. Lost Data Traffic Messages by Link

DATA ELEMENT	TYPE	JADS-AFEWES	JADS-ACETEF	AFEWES-ACETEF
Live Entity State (A/C TSPI)	UDP	Avg Lost: .08 Percent Lost: < 1 Max: 17	Avg Lost: 0 Percent Lost: 0 Max: 0	N/A
Threat Performance (Threat Track Data)	UDP	Avg Lost: .08 Percent Lost: < 1 Max: 17	Avg Lost: 0 Percent Lost: 0 Max: 0	N/A
Threat Performance (T/E, J/S, Target Location)	UDP	Avg Lost: .2 Percent Lost: < 1 Max: 36	N/A	Avg Lost: .002 Percent Lost: < 1 Max: 1
SUT_Jammer_Tech (DSM RF Emissions)	TCP	N/A	Avg Lost: .006 Percent Lost: <1 Max: 1	Avg Lost: .006 Percent Lost: <1 Max: 1
SUT_Receiver_Track (Verify Environment)	TCP	N/A	Avg Lost: 0 Percent Lost: 0 Max: 0	N/A
Source_Mode Change (Threat RF Emission)	TCP	Avg Lost: 0 Percent Lost: 0 Max: 0	N/A	Avg Lost: 0 Percent Lost: 0 Max: 0

A/C = aircraft T/E = tracking error

The two runs with UDP data losses were marked and further studied for anomalies before being included in the SUT valid data set, as was the run with the TCP data loss. The cause of the majority of the lost data was determined to be a short duration network equipment outage that impacted the JADS - AFEWES link for approximately one second. This same event contributed to some of the extreme data latency problems experienced. There was not conclusive evidence to identify the cause of the data loss event for the second run; only a single data packet was lost.

The circumstances surrounding the loss of the single reliable data message are far more interesting as TCP data traffic by its very nature should not have been lost. Analysis of the *EtherPeek* packet sniffer data for this event provided evidence to strongly suggest that the RTI itself was responsible for the lost ACETEF SUT_Jammer_Tech message to both AFEWES and JADS. The *EtherPeek* packet data collected at ACETEF confirmed that the message was transmitted; and the *EtherPeek* packet data collected at JADS and AFEWES showed that it was received by the RTI and acknowledged at both sites. However, it was not recorded by the loggers at either destination, indicating that for some reason the RTI simply did not pass the message content on to the federates. This run was an excursion run, during which four threat systems were active instead of two; yet, network tools did not indicate any problems or errors, or even excessive traffic levels, at the time the message was sent. The data from this excursion run were not included in the valid SUT data set.

For the second approach, analysts detected all data losses longer than 1 second and attempted to categorize the cause and outcome of each data loss using a combination of observer notes and test instrumentation. Since there were so few data loss events during Phase 3, this technique revealed

only one other strange data real-loss occurrence. On multiple occasions, best effort link health data messages, transmitted by ACETEF before the start of the run, were sent via different IP multicast groups, one of which did not forward the messages to its destination. Typically, the ACETEF federate was the last to join the federation on occasions where these losses occurred. There was no possibility of any SUT data impact from these link health message losses, but further study may indicate that anomalies in the federation joining process could cause best effort message traffic losses.

Router and RTI upgrades made prior to Phase 3 execution were given credit for the significant reduction in the amount of data lost in this phase over the previous phase.

Conclusions/Recommendations. Unlike Phase 2, where no TCP (reliable) data traffic losses were detected, one unique TCP reliable SUT_Jammer_Tech message was lost during Phase 3. JADS analysts attribute responsibility for the lost TCP message to the RTI itself, but the specific cause of the unusual RTI behavior could not be pinpointed. Less than 1% of all UDP (best effort) data traffic was lost, which were even fewer losses than in Phase 2. Of the two runs with UDP data losses that were marked for further study, neither run was actually excluded from the SUT valid data set. As with Phase 2 analysis, detailed analysis of each data loss event did not always result in successful determination of the cause of the event, although some data losses were obviously attributable to the known network link outage. Increased data packet instrumentation at each federate, via the use of *EtherPeek* data packet sniffers, did provide helpful detail for analysis into the cause of data loss events.

5.1.11 Measure 2-1-2-4. Average and peak data latency.

Intent. This measure was intended to report the average and peak latency experienced by Phase 3 test data elements while traversing the ADS network.

Data Collection and Analysis Approach. The RTI interface log files were used to record the arrival and departure times of all published and subscribed complex data types for each federate. JADS analysts utilized software tools to summarize and compare log file contents and determine round-trip federation latency as well as node-to-node latency values for particular complex data types. One additional tool, the *EtherPeek* data packet sniffer and analysis package, was added to the data collection and analysis tool set for Phase 3.

Data Sources. The RTI interface loggers at each federate recorded the arrival and departure times of all published and subscribed complex data types. Analysis of *EtherPeek* data post-test shed insight into the behavior of RTI software, federate software, and network in transporting data packets and helped analysts pinpoint the cause of anomalous latencies.

Results. The analysis of latency for Phase 3 was again approached in two ways; one focusing on node-to-node latency while the other focused on latency for just those federation messages deemed latency critical, (i.e., the time for MS_Source_Mode_Change, SUT_Jammer_Tech_Com, and SUT_Receiver_Track_Update messages to travel round trip between the AFEWES and ACETEF federates.)

For the first approach, node-to-node latency values across relevant network links were evaluated for six complex data message types. The six types were selected for evaluation based on their ability to provide insight into the impact of latent node-to-node traffic on SUT data validity. In other words, these were the message types that, if latent, should have had the most noticeable effect on SUT behavior and the SUT performance measure data collected. Eight individual runs with unusually high node-to-node latency values, out of 223 completed runs, were marked and further studied before being included in the SUT valid data set. Table 14 provides a summary of node-to-node latency categorized by message type and network link. For Phase 3 the highest node-to-node latency value detected was 1.5 seconds compared to values as high as 13 seconds detected during Phase 2.

Table 14. Node-to-Node Traffic Latency by Data Element (milliseconds)

DATA ELEMENT	TYPE	JADS-AFEWES	JADS-ACETEF	AFEWES-ACETEF
Live Entity State (A/C TSPI)	UDP	Avg: 45.7 Max: 1150	Avg: 44.4 Max: 472	N/A
Threat Performance (Threat Track Data)	UDP	Avg: 52.7 Max: 1151	Avg: 52.3 Max: 516	N/A
Threat Performance (T/E, J/S, Target Location)	UDP	Avg: 30.0 Max: 515	N/A	Avg: 41.2 Max: 511
SUT_Jammer _Tech (RF Emissions)	TCP	N/A	Avg: 75.3 Max: 312	Avg: 67.3 Max: 296
SUT_Receiver_Track (Verify Environment)	TCP	N/A	Avg: 77.0 Max: 267	N/A
Source_Mode Change (Threat RF Emission)	TCP	Avg: 55.7 Max: 372	Avg: 71.9 Max: 1548	Avg: 45.5 Max: 501

A/C = aircraft T/E = tracking error

For the second approach, analysts calculated round-trip federation latency values by summing the individual message latencies between AFEWES and ACETEF nodes for latency critical message types. Out of 223 successfully completed runs, only one run experienced unsuitable round-trip federation latency values (> 500 ms) and was marked for probable exclusion from the valid SUT data set. This showed a marked improvement over Phase 2 where eight runs experienced unsuitable round-trip federation latency. Calculated average and maximum round-trip federation latencies for Phase 3, based on the remaining successful runs, were 164 ms and 417 ms, respectively.

Further analysis was performed on the marked runs to determine the potential causes of the high latency values experienced. With the help of the *EtherPeek* sniffer data, several causes were identified. The first, also experienced during Phase 2, was a simple network link or network equipment outage. The impact of a short duration network link drop was a loss of best effort traffic and a reliable traffic delay. Although this was the most infrequent cause of latent traffic, it was responsible for the most extreme Phase 3 latency values. A second cause of unusual latencies was the inconsistent treatment of the execution control stop messages by the various federates

upon federation termination. On occasion, a federate, usually AFEWES, was able to continue sending data packets after the stop command was received, resulting in lost or latent messages to other federates. Stricter implementation of federation termination procedures, in accordance with the ICD, might alleviate this behavior. Third, incoming best effort messages were occasionally, inexplicably delayed by federation or RTI software after reaching the federate LAN. Message delivery problems within a federate were probably the second most frequent cause of latent data during Phase 3; the most frequent cause was the unusual behavior of the ACETEF reliable distributor in delivering outgoing messages to the other federates. Messages sent by the ACETEF reliable distributor experienced differential latencies to their destinations because of sequential outgoing packets being differentially held up at the sending end. No explanation could be found for this unusual reliable distributor behavior.

Conclusions/Recommendations. High round-trip federation and node-to-node latencies impacted a total of 9 out of 223 successfully completed test trials. Thus, approximately 4% of the trials had to be more carefully researched to determine if high latency resulted in SUT data anomalies, compared to 7% during Phase 2. Of these nine individual runs, no runs were actually excluded from the SUT valid data set. Detailed analysis of individual latency events resulted in the identification of an inconsistent treatment of the execution control stop messages by the various federates, and several within federate message delivery problems caused by some preemption of RTI processor control. Increased data packet instrumentation at each federate, via the use of *EtherPeek* data packet sniffers, did provide helpful detail for analysis into the cause of data latency events. Data latency can be managed through proper test design.

5.1.12 Measure 2-1-3-1. Degree to which test events (trials) were affected by ADS components (failure or otherwise) exclusive of network problems.

Intent. The intent of this measure was to determine the impact of ADS component availability on test trial events.

Data Collection and Analysis Approach. The HSNPL was used in conjunction with test control logs, event logs, and site notes to document ADS component problems and aborted runs. JADS analysts reviewed, categorized, and summarized the number, type, and duration of problems encountered to determine the number of test events (trials) impacted by ADS components.

Data Sources. The HSNPL was used to document ADS component problems, as were test control logs, event logs, and site observer notes. Information recorded in these written logs included notes on all problems experienced, problem start and stop time, and the particular trial events impacted.

Results. As in Phase 2, ADS component problems were encountered frequently during Phase 3 testing and were responsible for almost all the test time lost because of delays and failures (10 hours lost out of 48.8 hours total active test time.) The specific problems included federate crashes, real-time analysis tool (ADRS) crashes, AFEWES and ACETEF equipment software and

mechanical problems, data dropouts, and federation setup and script problems. Table 15 summarizes the impact of ADS component problems on trial events for Phase 3.

Table 15. Trials Lost to ADS Component Problems

Total test trials	270
Total aborted trials	32
ADS component problems faulted in aborted trial	21
Federate and analysis tool crashes	3
Federation setup and script problems	10
Communication tool and data dropouts	1
ACETEF mechanical and software problems	7
AFEWES mechanical and software problems	10

Figure 13 shows the breakdown of aborted trial runs by fault category. ADS component problems were responsible for 21 lost runs (66 %) during Phase 3 testing.

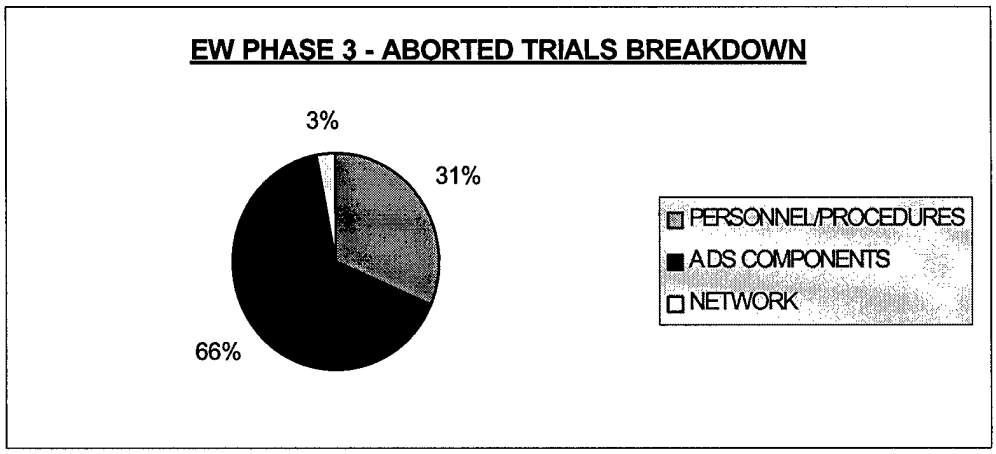


Figure 13. Phase 3 Aborted Trials Breakdown

Figure 14 shows lost runs because of ADS component problems further separated into two categories in order to provide additional insight into the impacts of existing test equipment failures versus failures attributable to system hardware and software implemented specifically to enable distributed testing (e.g., RTI software.)

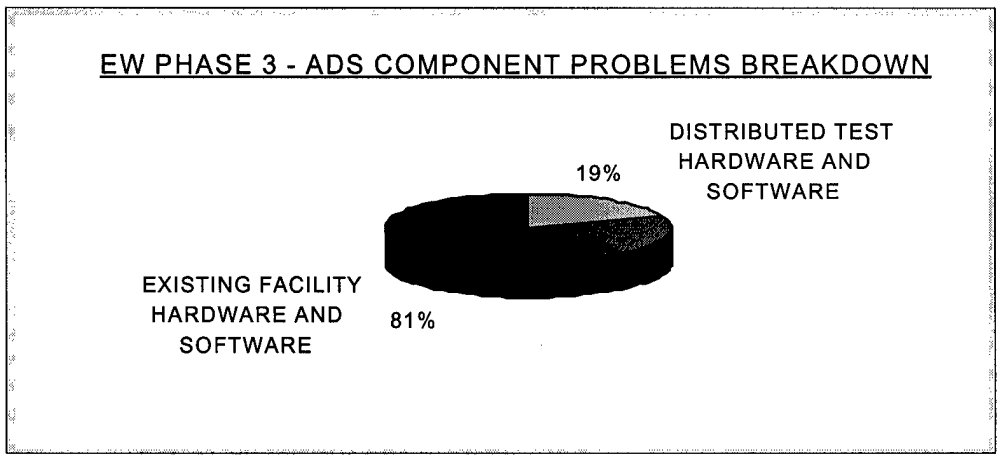


Figure 14. ADS Component Problems Breakdown

Conclusions/Recommendations. ADS component problems, such as federate crashes, real-time analysis tool (ADRS) crashes, AFEWES and ACETEF equipment problems, data dropouts, and federation setup and script problems were responsible for the majority of lost test time in Phase 3 and resulted in 21 aborted runs. Approximately 19% of these aborted runs (4 runs) were lost because of problems with equipment implemented to enable distributed testing. The remaining 81% (17 runs) were lost because of problems with existing facility equipment (primarily at ACETEF), although, admittedly, some of the ACETEF systems were used nontraditionally for Phase 3 testing.

5.1.13 Measure 2-1-3-2. Degree to which test events (trials) were affected by network problems (failure or otherwise).

Intent. The intent of this measure was to determine the impact of ADS network availability on test trial events. The network system included all software and hardware used for connecting the distributed sites between routers.

Data Collection and Analysis Approach. The HSNPL was used in conjunction with test control logs, event logs, and site notes to document ADS network problems and aborted runs. JADS analysts reviewed, categorized, and summarized the number, type, and duration of problems encountered to determine the number of test events (trials) impacted by the ADS network.

Data Sources. The HSNPL was used to document ADS network problems, as were test control logs, event logs, and site observer notes. Information recorded in these written logs included notes on all problems experienced, problem start and stop times, and the particular trial events impacted.

Results. Network problems were nearly nonexistent during Phase 3 testing and resulted in just one aborted trial; the JADS-AFEWES link was lost for approximately one minute, resulting in a loss of TSPI data at AFEWES. A second JADS-AFEWES link problem, the result of a

cryptographic equipment glitch, caused two-way losses of best-effort (UDP) data and a delay of reliable (TCP) traffic but was not of significant duration (only .9 seconds) to result in anomalous SUT data or abortion of the trial. This event is discussed further under lost data and latency measures sections. Table 16 summarizes the impact of ADS network problems on trial events.

Table 16. Trials Lost Because of ADS Network Problems

TOTAL TEST TRIALS	270
TOTAL ABORTED TRIALS	32
ADS NETWORK PROBLEMS FAULTED	1

Figure 15 shows the breakdown of aborted trial runs by fault category. ADS network problems were responsible for 3% of the lost runs during Phase 3 testing.

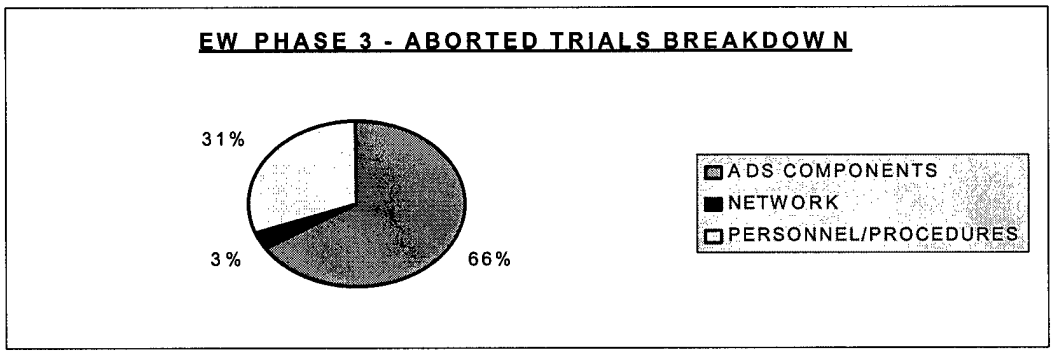


Figure 15. Aborted Trials Breakdown

Conclusions/Recommendations. ADS network problems, practically nonexistent in Phase 3, were responsible for just a minute or two of lost test time and resulted in only one aborted run.

5.1.14 Measure 2-1-3-3. Degree to which test events (trials) were affected by personnel problems.

Intent. This measure was intended to identify the impacts of personnel problems including problems related to training, manning, consistency, and coordination on test trial events. This measure was intended to collect data on the human element of an ADS test and the impacts associated with human error and human creativity.

Data Collection and Analysis Approach. The HSNPL was used in conjunction with test control logs, event logs, and site notes to document personnel and procedural problems and aborted runs. JADS analysts reviewed, categorized, and summarized the number, type, and duration of problems encountered to determine the number of test events (trials) impacted by personnel.

Data Sources. The HSNPL was used to document problems encountered with personnel and procedures, as were test control logs, event logs, and site observer notes. Information recorded in these written logs included notes on all problems experienced, problem start and stop times, and the particular trial events impacted.

Results. Personnel and procedural problems experienced during Phase 3 testing including script problems, miscommunication among test controllers and test station operators, and other miscellaneous operator errors accounted for a much higher percentage (33%) of lost trials than in Phase 2 (11%). Simple mistakes, such as inputting an accurate script number or incorrectly setting a system switch, were made most probably because of lack of concentration. Table 17 summarizes the impact of ADS component problems on trial events.

Table 17. Trials Lost Because of Personnel and Procedural Problems

TOTAL TEST TRIALS	270
TOTAL ABORTED TRIALS	32
PERSONNEL/PROCEDURES FAULTED	10

Figure 16 shows the breakdown of aborted trial runs by fault category. Personnel and procedural problems were responsible for 33% of the lost runs during Phase 3 testing.

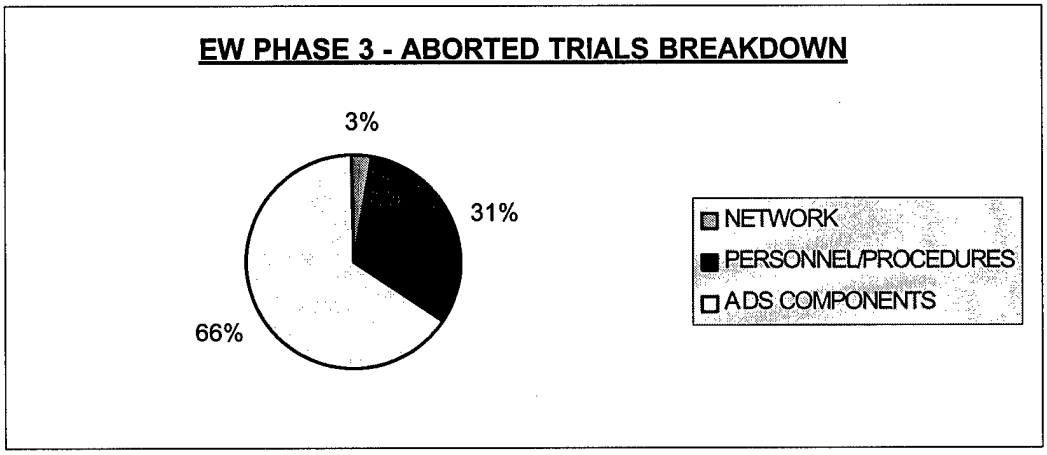


Figure 16. EW Test Phase 3 Aborted Trials Breakdown

Conclusions/Recommendations. Problems relating to personnel and procedures during Phase 3 execution consisted almost entirely of operator error based primarily on miscommunication or simple lack of concentration. Most of these problems, which were responsible for the second largest portion of lost test time, could probably have been avoided.

5.1.15 Measure 2-2-1-4. Ease with which data can be retrieved, post-trial, from a given node.

Intent. The intent of this measure was to determine the degree of difficulty in retrieving ADS data from the distributed EW Test nodes.

Data Collection and Analysis Approach. Interviews were conducted with Phase 3 data managers and analysts to ascertain their perceptions about ADS post-trial data retrieval. Interview questions focused on identifying particular problems noted using retrieval procedures or tools. JADS analysts reviewed and summarized recorded remarks, also noting differences between Phase 2 and Phase 3 where appropriate.

Data Sources. Seven interviews were conducted. Interviewees included the TCAC data manager, the AFEWES and ACETEF data management representatives, and four JADS data analysts.

Results. Phase 3 interviewees noted an improvement in the coordination of data retrieval over Phase 2, but the methods and tools that were used for bringing distributed site data to the TCAC for analysis remained the same. Methods ranged from handcarrying written observer logs to the quick and easy electronic transfer of large data files using UNIX file transfer protocols. The federate log files and other system data created at distributed sites were typically transferred to the TCAC in less than 30 minutes. Although usually the data were transferred after test completion each day, some site data (especially that generated by the newly implemented *EtherPeek* tool) were transferred during breaks to prevent an end-of-day data backlog, as well as to demonstrate the ease of data transfer. Most respondents agreed that the data retrieval procedures were more formally documented and briefed to all participants for Phase 3, based on recommendations from Phase 2, for better coordination. In Phase 2 some frustration had occurred between distributed site data managers and the data analysts because of miscommunication about when data would be sent and where they would be stored. Clear documentation as to procedures and storage directory and file structures alleviated this frustration for Phase 3, although several respondents admitted that even more improvement could be made, as data collection during Phase 3 was not flawless. In fact, several site log files had to be retransmitted after Phase 3 execution was complete, and one test day of *EtherPeek* data were never recovered. On a side note, electronic file transfer of data increases utilized network bandwidth to near 100%, a level that would indicate a problem during typical testing. Better documentation of file transfer events and associated times would make network load monitoring and analysis an easier task.

Conclusions/Recommendations. In general, respondents indicated that while the data retrieval methods and tools implemented for Phase 3 testing were no different than those implemented for Phase 2, the coordination of data retrieval for Phase 3 was much improved. Improvements, based on recommendations from Phase 2 included better documentation on when data should be sent and where they should be stored, as well as better dissemination of expected procedures to all participants. Interviewees admitted that the implementation of an even more well-defined directory structure for organizing the retrieved data might have prevented the few data losses that did occur.

It should be noted that it took up to 10 days after traditional test events in Phase 1 to obtain the electronic data from the test facility.

5.1.16 Measure 2-2-2-1. Degree to which test managers can control the configurations of ADS participants, the ADS environment data, and ADS networks.

Intent. This measure was intended to assess the ability of the test manager to adequately control the configuration of ADS participants, the ADS environment data, and ADS networks both during and between test events.

Data Collection and Analysis Approach. Interviews were conducted with the EW Test manager and test controllers to ascertain their perceptions about ADS configuration control. Interview questions required assessment of documents, tools, and reports used to establish configuration control as well as the overall effectiveness of configuration control procedures. Interviewees were asked to make recommendations for improvement to the configuration control process. JADS analysts reviewed and summarized recorded remarks, also noting differences between Phase 2 and Phase 3 where appropriate.

Data Sources. Four interviews were conducted. Interviewees included the EW Test manager, the JADS test controllers, and the ACETEF and AFEWES test executors.

Results. The general consensus among respondents was that configuration control of ADS component software, networks, and environment data prior to and during testing, although an improvement over that experienced during Phase 2, was still limited. There were some procedures in place, such as usable script lists and software library lists, but the EW Test manager was not in direct control over the software and system versions implemented at ACETEF or AFEWES nodes. These were managed at the site level with, supposed, test manager awareness. However, many of the problems caused by constantly changing contractor test component software and test tools that plagued Phase 2 integration, acceptance testing, and test execution were not experienced during Phase 3. One can suppose this was because Phase 2 execution served to work out any remaining bugs in federation hardware and software that were reused in Phase 3 testing. A new configuration issue seen in Phase 3 was the lack of control over the level of training of ADS test participants. Phase 3 testing required that the AFEWES threat system operators be trained at a level similar to those who participated in Phase 2. The employment of some less experienced operators during Phase 3 resulted in the post-test deletion of some collected SUT data. Although configurations were certainly stable enough to permit successful Phase 3 test execution, most respondents did not feel comfortable with the level of configuration control obtained and suggested that more top-down procedures and written documentation be kept for maintaining stricter configuration control. The network under direct JADS control did not pose any configuration problems.

Conclusions/Recommendations. Although fewer configuration control problems were experienced during Phase 3, respondents in general did not feel that configuration control procedures were satisfactory. Recommendations included allocating more control to the test manager and formalizing documentation and procedures utilized by contracted support personnel.

5.1.17 Measure 2-3-2-3. Degree to which protocols, processes, and procedures are needed to enable effective centralized test control.

Intent. This measure was intended to determine the degree to which protocols, processes, and procedures were needed to enable effective centralized test control.

Data Collection and Analysis Approach. Interviews were conducted with the Phase 3 test controllers and test executors to ascertain their perceptions about effective centralized test control for ADS. Interview questions focused on the effectiveness of documented test control procedures, as well as the need for modifying and adding procedures during test. Respondents were asked to identify particular areas of difficulty and potential fixes. JADS analysts reviewed and summarized recorded remarks, also noting differences between Phase 2 and Phase 3 where appropriate.

Data Sources. Seven interviews were conducted. Interviewees included the JADS test controller positioned in the TCAC, three JADS test executors positioned at AFEWES and ACETEF, and three test station operators positioned in the TCAC.

Results. According to interview responses, Phase 3 demonstrated marked improvement over Phase 2 in terms of effective centralized test control. Although there were still some difficulties experienced with the volume and clarity of voice communication, all interviewed personnel indicated strong feelings of effective centralized test control for Phase 3, including the test executors at the distributed sites who had often felt out of the loop during Phase 2. On a 1-6 scale, representing the spectrum from ineffective centralized control to effective centralized control, all Phase 3 respondents provided ratings of 5 and 6, whereas two distributed respondents provided ratings of 3 and 4 during Phase 2. Most interviewees still mentioned having difficulty hearing voices from the distributed locations and having to ask for clarification on numerous occasions when test control instructions could not be clearly distinguished. However, due to fairly strict adherence to structured control processes and procedures, this was usually not a problem that resulted in trial losses. Faulty communications were mainly distracting when site observers attempted to discuss malfunctioning federation equipment with the test controller or executors at distributed sites. One respondent discussed the difficulty in maintaining control in the TCAC when test station operators became tired and lost concentration. Multiple runs were lost during Phase 3 because of execution errors that could have been alleviated if personnel had been concentrating more fully on an assigned task.

Conclusions/Recommendations. In general, respondents seemed to feel that the protocols, processes, and procedures implemented for Phase 3 enabled effective centralized test control to take place. Moreover, better defined communications processes for Phase 3 helped avoid the trial losses caused by miscommunication that plagued the Phase 2 execution. Interviewed personnel discussed several recommendations for improved test control protocols, processes, and procedures.

1) The addition of an extra voice communications line would have enabled distributed personnel to discuss non-test-dependent federation equipment problems without delaying further test execution.
2) An unusually high number of trial losses due to personnel errors might indicate a reduced level of test station operator concentration because of fatigue.
3) Common displays across distributed sites might enable better understanding of federation and system status by all participants and therefore eliminate the need for extenuated voice conversation.

5.1.18 Measure 2-3-2-4. Degree to which real-time analysis systems support test safety and other test control requirements.

Intent. This measure was intended to determine what real-time analysis was required and the impact of having real-time analysis systems for test control.

Data Collection and Analysis Approach. Interviews were conducted with the Phase 3 test controllers and test executors to ascertain their perceptions about having real-time analysis systems for test control. Interview questions focused on the manner in which real-time analysis tools provided feedback that improved the test controller's ability to control test events. JADS analysts reviewed and summarized recorded remarks, also noting differences between Phase 2 and Phase 3 where appropriate.

Data Sources. Seven interviews were conducted. Interviewees included the JADS test controller positioned in the TCAC, three JADS test executors positioned at AFEWES and ACETEF, and three test station operators positioned in the TCAC.

Results. Interview responses from Phase 3 were almost identical in nature to those from Phase 2. There was fairly unanimous agreement that the feedback provided by real-time analysis tools did improve the ability of test controllers to control test events. Such tools allowed test controllers to watch distributed events unfold according to expectation from a central facility and gave nearly immediate identification of distributed component problems and data losses. They provided a cursory feel for the usefulness of each test trial and enabled test controllers to make timely decisions about trial events or system problems. The only negative feedback from Phase 3 was that these tools were not sufficient to detect all important aspects of federation behavior in a timely manner. In particular, there were several time synchronization losses that occurred during Phase 3. Most of these losses were detected during test execution by highly observant test executors, although not immediately or during post-test analysis. If they had not been detected real time or had been more severe, these problems could have resulted in hours of wasted test time spent acquiring invalid SUT data until the next scheduled time synchronization check was performed.

Conclusions/Recommendations. Real-time analysis tools greatly enhanced the ability of ADS test controllers to control test events and provided greater situational awareness to other test participants. It is still important, however, to consider what areas of test control (e.g., time synchronization) may not be covered by real-time tools and to implement procedures which

ensure that any problems do not go undetected for long enough to cause severe impacts to testing efficiency. No conclusion was made about the ability of such tools to impact test safety, as there were no safety issues identified for the EW Phase 3 test.

6.0 Correlation Analysis

The main emphasis of EW testing for both Phase 2 and Phase 3 was on the impact of ADS on the test components rather than on the performance of the test item itself. EW Test MOPS were computed, not only for their utility in determining SUT and threat performance but as a means of determining potential ADS impacts. Differences between data sets collected during ADS testing and baseline data collected by more traditional test means in the OAR and HITL phases were hoped to point analysts to areas where ADS testing may have significant impacts; likewise, similarities between ADS and non-ADS phase data sets were to confirm little ADS impact. It was expected that Phase 2 and Phase 3 MOP data would not differ significantly from baseline data for most threat systems and reference test conditions. Correlation analysis, via the use of statistical hypothesis testing, was aimed at performing pair-wise comparisons of data sets across phases for each threat system and reference test condition, so that any differences could be identified and possibly attributed to ADS causes. However, the unexpectedly large influences on the collected data from all EW Test phases because of operator variance and differing threat system representations between facilities hampered the ability of analysts to clearly realize the impacts of ADS, sometimes skewing data sets in extraordinary ways.

6.1 EW Test Measure of Performance (MOP) Evaluation

During Phase 3 testing, as for Phase 2, SUT performance data were collected for the ten EW Test measures listed in Table 1. The evaluation process for each MOP included sorting the collected raw data into sets by threat system and reference test condition, determining the distribution shape and parameters for each data set, calculating descriptive statistics, and correlating each Phase 3 data set to the OAR and HITL baseline data, as well as to the Phase 2 data. All Phase 3 classified descriptive statistics and frequency histograms showing the shape, central tendency, and dispersion or variance of the MOP data sets are published in the *EW Test Classified Results Report*. The following sections detail the correlation process and the results of the correlation analysis that was performed between matching data sets (i.e., data collected during different EW Test phases under the same reference test condition) using statistical hypothesis testing.

6.2 Statistical Hypothesis Testing

Two sample data sets were said to correlate or equate if it could be determined that they were analogous with respect to certain distribution parameters. Statistical hypothesis testing provided a means of determining how well the distribution shape, location, and dispersion parameters of two data sets equated. For each distribution parameter, an appropriate statistical comparison test was selected based on the distribution form of the collected data (e.g., binomial, normal). The underlying hypothesis of each test was that the two data samples were equivalent; that is, that they represented the same true population. If, in performing the test, this hypothesis could not be rejected with reasonable confidence, then the two data sets were determined to correlate.

Numerous statistical hypothesis tests were available for comparing the distribution parameters of sample data sets. However, valid application of any one of these tests necessitated meeting its

underlying assumptions and data requirements. For example, some tests were invalid when applied to nonnormal or noncontinuous data; others required a minimum number of samples to provide power in distinguishing between data sets. Successful application rested in choosing a test that was both valid and powerful in determining the extent to which the data sets correlated. Four comparison tests with wide utility were chosen for application in MOP correlation analysis after the characteristics of the data collected during initial EW Test phases were assessed. These included a comparison of proportions test applicable to binomially distributed (pass/fail) data; a T-test or means comparison test for comparing location parameters for roughly normal shaped data; an F-test or variance test for comparing the dispersion of roughly normal shaped data; and a Kolmogorov-Smirnov (K-S) test for correlating the overall shape including the mean and variance of data sets that may not meet an underlying assumption of normality.

Regardless of the type of hypothesis test selected, the methodology for performing the test was standard. For each, a test statistic value was computed by inserting the collected data values from the two samples into a mathematical equation. However, it was not possible, using statistical hypothesis testing, to conclude that two sets of sample data were the same with absolute certainty. Instead, the underlying hypothesis must be accepted or rejected based on the probability of obtaining the generated test statistic, along with the tester's willingness to risk making an incorrect conclusion. If the underlying hypothesis was true, and the two data sets were indeed from the same population, it was highly probable that the test statistic value generated fell within an expected range. If it fell outside this range, (i.e., is more extreme), then it was more likely that the two samples were not analogous for that distribution parameter. For each type of hypothesis test, there was a statistical table that associated a probability value or P-value with the generated test statistic value based on the range of values expected when the underlying hypothesis was true. This P-value was the result value reported, (e.g., $p = .0452$). In statistical terms, the P-value represented the probability of attaining the given value of the test statistic or a more extreme value if the null hypothesis was true. To use the P-value, it must be compared to the level of risk the tester was willing to take in incorrectly rejecting the underlying assumption. Essentially, the underlying hypothesis should only be rejected if the tester was comfortable with a level of risk greater than the P-value reported. If the tester is only willing to risk a 5% chance of an incorrect rejection, then for any P-value $< .05$, the underlying hypothesis should be rejected. If a 10% risk is acceptable, the underlying hypothesis may be rejected for P-values up to 0.10.

6.3 Correlation Results

This section contains the results of the correlation tests performed on each of the MOP data sets. Each available data set within a reference test condition (north dry, north wet, south dry and south wet) was correlated against every other data set. Thus, for example, the Phase 3 data set for missile miss distance was correlated with the Phase 2, HITL, and OAR data sets. The resultant P-values from each of the three correlation tests (T-test, F-test, and K-S test) are shown in tables 18 through 43.

To reference each EW Test phase, use the following abbreviations:

P3 - data set collected using Phase 3 data - latency is included where appropriate in all measurements
P3L - response time data collected during Phase 3 with latency removed
P2 - data set collected using Phase 2 data - latency is included where appropriate in all measurements
P2L - response time data collected during Phase 2 with latency removed
HITL - data set collected using HITL data prepared by GTRI - latency is included where appropriate in all measurements
OAR - data set collected using OAR data prepared by GTRI - latency is included where appropriate in all measurements

When referencing the columns in the following tables, the two labels shown are the two data sets compared in that correlation test. For example, the column reading P3-P2 shows the correlation results for the Phase 3 to Phase 2 comparison. P3-P2 provides the same results as P2-P3, so all possible correlation combinations are shown.

Details about each MOP, including a basic description, purpose, instrumentation facts, analysis methodology, summary statistics, and frequency histograms, are published in the *EW Test Classified Results Report.*

6.3.1 Correct ID Response Time

Table 18. System 1 Correct ID Response Time Correlation Matrix

Test	P3L-P3	P3L-P2L	P3L-P2	P3L-SIL	P3-P2L	P3-P2	P3-SIL	P2L-P2	P2L-SIL	P2-SIL
North Wet										
T-test	.0024	*	.0020	.2211	*	.9355	.0000	*	*	.0000
F-test	.4663	*	.0000	.3783	*	.0000	.3428	*	*	.0000
K-S Test	.0638	.0000	.0000	.9932	.0000	.0218	.0080	.0000	.0000	.0000
South Wet										
T-test	.0012	.8242	.0000	.8104	.0000	.6496	.0001	.0000	.4683	.0000
F-test	.4830	.0000	.0000	.4411	.0000	.0000	.4564	.0000	.0000	.0000
K-S Test	.0516	.0071	.0000	1.000	.0000	.0186	.0061	.0000	.0001	.0000

NOTE: *Phase 2L southbound wet data had no variance, so P-value could not be computed.

Table 19. System 2 Correct ID Response Time Correlation Matrix

Test	P3L-P3	P3L-P2L	P3L-P2	P3L-SIL	P3-P2L	P3-P2	P3-SIL	P2L-P2	P2L-SIL	P2-SIL
North Wet										
T-test	.0951	.0000	.5352	.0254	.0000	.0919	.0000	.0000	.0590	.0000
F-test	.4600	.0000	.0000	.2593	.0000	.0000	.3216	.0000	.0000	.0000
K-S Test	.5210	.0000	.0009	.2655	.0000	.0025	.0206	.0000	.0000	.0000
South Wet										
T-test	.0364	.0362	.4087	.0168	.0000	.0004	.0000	.0002	.2999	.0049
F-test	.4979	.0000	.0000	.2845	.0000	.0000	.2768	.2918	.0000	.0000
K-S Test	.2402	.0001	.0173	.1127	.0000	.0001	.0057	.0000	.0007	.0003

Table 20. System 3 Correct ID Response Time Correlation Matrix

Test	P3L-P3	P3L-P2L	P3L-P2	P3L-SIL	P3-P2L	P3-P2	P3-SIL	P2L-P2	P2L-SIL	P2-SIL
North Wet										
T-test	.5308	.4931	.9011	.9039	.0690	.2529	.3059	.0000	.2044	.9879
F-test	.3698	.0000	.0000	.0018	.0000	.0000	.0029	.1022	.0000	.0000
K-S Test	.3773	.0083	.0010	.4834	.0005	.0013	.8631	.0013	.0001	.0004
South Wet										
T-test	.4071	*	.2078	.6466	*	.9185	.8024	*	*	.6962
F-test	.4826	*	.0000	.0004	*	.0000	.0005	*	*	.0000
K-S Test	.9791	.0005	.0010	1.000	.0002	.0000	1.000	.0000	.0000	.0000

NOTE: *Phase 2L southbound wet data had no variance, so P-value could not be computed.

Table 21. System 4 Correct ID Response Time Correlation Matrix

Test	P3L-P3	P3L-P2L	P3L-P2	P3L-SIL	P3-P2L	P3-P2	P3-SIL	P2L-P2	P2L-SIL	P2-SIL
North Wet										
T-test	.5738	.9006	.5588	.5661	.4525	.9015	.1187	.1141	.1334	.0134
F-test	.4236	.0000	.0000	.0567	.0000	.0000	.0478	.5041	.0000	.0000
K-S Test	1.000	.0341	.0264	1.000	.0750	.0063	.3731	.0167	.0020	.0001

6.3.2 Correct ECM Technique Selection Response Time

Table 22. System 1 Correct ECM Technique Selection Response Time Correlation Matrix

Test	P3L-P3	P3L-P2L	P3L-P2	P3L-SIL	P3-P2L	P3-P2	P3-SIL	P2L-P2	P2L-SIL	P2-SIL
North Wet										
T-test	.0029	*	.0005	.0010	*	.7279	.0000	*	*	.0000
F-test	.4657	*	.0000	.0061	*	.0000	.0069	*	*	.0000
K-S Test	.1021	.0000	.0077	.0232	.0000	.0644	.0001	.0000	.0000	.0000
South Wet										
T-test	.0210	.9613	.0446	.0491	.0084	.5713	.0000	.0170	.0163	.0000
F-test	.3017	.3732	.4327	.0003	.1756	.2214	.0000	.4312	.0004	.0002
K-S Test	.0804	.0136	.0136	.5542	.0000	.0408	.0002	.0000	.0001	.0000

NOTE: Phase 2L northbound wet data had no variance, so P-value could not be computed.

Table 23. System 2 Correct ECM Technique Selection Response Time Correlation Matrix

Test	P3L-P3	P3L-P2L	P3L-P2	P3L-SIL	P3-P2L	P3-P2	P3-SIL	P2L-P2	P2L-SIL	P2-SIL
North Wet										
T-test	.0949	.0000	.6004	.0166	.0000	.0750	.0000	.0000	.0550	.0000
F-test	.4562	.0000	.0000	.3766	.0000	.0000	.3295	.0000	.0000	.0000
K-S Test	.6039	.0000	.0189	.3235	.0000	.0025	.0206	.0000	.0000	.0000
South Wet										
T-test	.0335	.0000	.0378	.0006	.0000	.0000	.0000	.0000	.0000	.0957
F-test	.4764	.2314	.2777	.2986	.2033	.2944	.2702	.0636	.4085	.1037
K-S Test	.3440	.0000	.4330	.0227	.0000	.1019	.0005	.0000	.0000	.0004

Table 24. System 3 Correct ECM Technique Selection Response Time Correlation Matrix

Test	P3L-P3	P3L-P2L	P3L-P2	P3L-SIL	P3-P2L	P3-P2	P3-SIL	P2L-P2	P2L-SIL	P2-SIL
North Wet										
T-test	.6104	.4772	.8818	.6407	.0786	.3010	.1895	.0000	.6427	.4194
F-test	.2403	.0000	.0000	.0026	.0000	.0000	.0139	.1030	.0000	.0000
K-S Test	.5391	.0032	.0010	.7351	.0054	.0015	.1321	.0003	.0008	.0004
South Wet										
T-test	.1992	*	.2317	.9306	*	.3418	.2005	*	*	.3832
F-test	.0000	*	.0000	.0003	*	.0000	.0002	*	*	.0000
K-S Test	.7150	.0005	.0024	.6713	.0000	.0009	.2872	.0000	.0000	.0000

NOTE: Phase 2L southbound wet data had no variance, so P-value could not be computed.

6.3.3 RMS Tracking Error

Table 25. System 1 RMS Tracking Error Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Dry						
T-test	.1608	.4435	.2163	.1474	.2179	.1024
F-test	.0000	.0209	.0045	.0000	.0000	.0000
K-S Test	*	*	*	*	*	*
South Dry						
T-test	.3003	.2908	.4660	.5092	.3137	.2804
F-test	.0000	.0000	.0001	.4124	.0000	.0000
K-S Test	*	*	*	*	*	*
North Wet						
T-test	.3812	.0001	.0015	.0017	.0022	.0215
F-test	.0000	.0000	.0000	.0000	.0000	.0000
K-S Test	.9513	.0014	.0000	.0000	.0000	.0000
South Wet						
T-test	.0290	.0003	.0000	.0000	.0000	.1572
F-test	.0000	.0000	.0000	.0000	.0000	.3607
K-S Test	.0000	.0000	.0000	.0000	.0000	.0173

NOTE:* OAR, HITL, Phase 2, and Phase 3 north dry, and OAR, HITL, and Phase 3 south dry data sets have less than 16 samples, so P-value is not computed.

Table 26. System 2 RMS Tracking Error Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Dry						
T-test	.9432	.8727	.6023	.8025	.5594	.6691
F-test	.3581	.4121	.0115	.4444	.0030	.0054
K-S Test	*	*	*	*	*	*
South Dry						
T-test	.6264	.2175	.6514	.5960	.9116	.3965
F-test	.2247	.0116	.3205	.0015	.1029	.0288
K-S Test	*	*	*	*	1.000	*
North Wet						
T-test	.0109	.4217	.8583	.0465	.0404	.6117
F-test	.3014	.3552	.0056	.4333	.0005	.0007
K-S Test	.2896	1.000	.9551	.1880	.5978	.8981
South Wet						
T-test	.1102	.0944	.0101	.9807	.2052	.1886
F-test	.0779	.0254	.2956	.2968	.1544	.0547
K-S Test	1.000	.3906	.0176	.9386	.0716	.0030

NOTE:* OAR, HITL, Phase 2, and Phase 3 north dry, and HITL, and Phase 3 south dry data sets have less than 16 samples, so P-value is not computed.

Table 27. System 3 RMS Tracking Error Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Dry						
T-test	.7563	.0000	.1562	.0000	.1222	.0000
F-test	.2662	.0001	.4252	.0000	.3479	.0001
K-S Test	1.000	*	*	*	*	*
South Dry						
T-test	.3946	.3826	.4580	.7421	.0335	.0515
F-test	.0000	.0000	.0000	.0559	.3087	.0159
K-S Test	*	*	*	*	*	.0088
North Wet						
T-test	.0534	.0209	.1697	.0014	.2566	.0373
F-test	.0000	.0000	.0000	.0000	.0000	.0000
K-S Test	1.000	.0002	.0021	.0004	.0045	1.000
South Wet						
T-test	.0202	.1790	.0044	.0743	.2209	.0116
F-test	.0000	.0000	.0000	.0000	.0007	.0000
K-S Test	.0105	1.000	.0000	.0170	.0003	.0000

NOTE:* OAR and HITL north dry, and Phase 2, and Phase 3 south dry data sets have less than 16 samples, so P-value is not computed.

Table 28. System 4 RMS Tracking Error Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Dry						
T-test	.6555	.0002	.0003	.0844	.0030	.0000
F-test	.0000	.4711	.0000	.0000	.2400	.0000
K-S Test	*	*	*	*	.0000	*
North Wet						
T-test	.0336	.0040	.2381	.0008	.0176	.0060
F-test	.0000	.0000	.0003	.0000	.0000	.0000
K-S Test	.3383	.0000	.0963	.0001	.0044	.0000

NOTE:* HITl and Phase 3 north dry data sets have less than 16 samples, so P-value is not computed.

6.3.4 Jamming-to-Signal Ratio

Table 29. System 1 Jamming-to-Signal Ratio Correlation Matrix

Test	P3-P2	P3-HITL	P2-HITL
North Wet			
T-test	.0000	.0000	.0000
F-test	.0000	.0000	.0000
K-S Test	.0000	.0000	.0000
South Wet			
T-test	.0000	.0000	.0000
F-test	.0000	.0000	.0000
K-S Test	.0000	.0000	.0000

Table 30. System 2 Jamming-to-Signal Ratio Correlation Matrix

Test	P3-P2	P3-HITL	P2-HITL
North Wet			
T-test	.0000	.0000	.0000
F-test	.0000	.0000	.0000
K-S Test	.0000	.0000	.0000
South Wet			
T-test	.0000	.0000	.0000
F-test	.0000	.0000	.0000
K-S Test	.0000	.0000	.0000

Table 31. System 3 Jamming-to-Signal Ratio Correlation Matrix

Test	P3-P2	P3-HITL	P2-HITL
North Wet			
T-test	.0000	.0000	.0000
F-test	.0000	.0000	.0000
K-S Test	.0000	.0000	.0000
South Wet			
T-test	.0000	.0000	.0000
F-test	.0000	.0000	.0000
K-S Test	.0000	.0000	.0000

Table 32. System 4 Jamming-to-Signal Ratio Correlation Matrix

Test	P3-P2	P3-HITL	P2-HITL
North Wet			
T-test	.0000	.0000	.0000
F-test	.0000	.0000	.0000
K-S Test	.0000	.0000	.0000

6.3.5 Number of Breaklocks

Table 33. System 1 Number of Breaklocks Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Dry						
T-test	.4344	*	.0736	*	.2874	*
F-test	.0003	*	.0000	*	.2256	*
K-S Test	**	**	**	**	**	**
South Dry						
T-test	.8810	*	.0437	*	.0409	*
F-test	.1908	*	.0270	*	.1193	*
K-S Test	**	**	**	**	*	**
North Wet						
T-test	.3587	.0009	.0000	.0258	.0000	.0000
F-test	.0019	.0001	.0000	.1530	.0000	.0000
K-S Test	1.000	.0000	.0000	.0663	.0000	.0000
South Wet						
T-test	.3427	.0103	.0000	.0005	.0000	.0000
F-test	.0068	.0006	.0000	.0000	.0000	.0000
K-S Test	1.000	.1344	.0000	.0177	.0000	.0000

NOTE: *HITL north dry, HITL south dry and Phase 2 north wet data sets have no standard deviation, so the T-test and F-test do not provide P-values.
** OAR, HITL, Phase 2, and Phase 3 north dry, and OAR, HITL, and Phase 3 south dry data sets have less than 16 samples, so P-value is not computed.

Table 34. System 3 Number of Breaklocks Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Dry						
T-test	.3432	.5385	.2206	.1792	.9095	.0953
F-test	.0243	.1898	.1314	.0078	.2215	.0396
K-S Test	**	**	**	**	**	**
South Dry						
T-test	.2515	.2976	.9573	.9206	.0085	.2614
F-test	.0310	.0631	.4513	.3947	.0312	.0659
K-S Test	**	**	**	**	**	**
North Wet						
T-test	.8650	.0134	.5358	.0038	.3635	.0004
F-test	.0133	.0000	.0839	.0005	.1939	.0000
K-S Test	1.000	.3958	.0000	.1185	1.000	.0168
South Wet						
T-test	.5075	.0276	.0261	.0016	.0834	.0000
F-test	.4552	.0001	.0141	.0000	.0125	.0000
K-S Test	1.000	.4351	.2627	.0588	.8446	.0006

NOTE: ** OAR, HITL, and Phase 2 north dry, and HITL, Phase 2 and Phase 3 south dry data sets have less than 16 samples, so P-value is not computed.

Table 35. System 4 Number of Breaklocks Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Dry						
T-test	*	*	*	.8787	.0001	.0001
F-test	*	*	*	.4562	.0003	.0026
K-S Test	**	**	**	**	**	**
North Wet						
T-test	.0001	.0567	.6147	.0048	.0036	.2647
F-test	.0013	.0445	.2815	.0868	.0002	.0123
K-S Test	.0116	.3176	1.000	.0050	.0406	.5884

NOTE: *Phase 3 north dry data set has no standard deviation, so the T-test and F-test do not provide P-values.
** HITL, Phase 2, and Phase 3 north dry data sets have less than 16 samples, so P-value is not computed.

6.3.6 Reduction in Engagement Time

Table 36. System 1 Reduction in Engagement Time Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Wet						
T-test	.1786	.0000	.0000	.0000	.0000	.0000
F-test	.0000	.0000	.0000	.1161	.0000	.0000
K-S Test	.4974	.0000	.0000	.0000	.0000	.0000
South Wet						
T-test	.2222	.0005	.0000	.0000	.0000	.0000
F-test	.0029	.3464	.0000	.0054	.0000	.0000
K-S Test	.0000	.0000	.0000	.0004	.0000	.0022

Table 37. System 3 Reduction in Engagement Time Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Wet						
T-test	.6862	.0434	.4945	.0000	.2494	.0168
F-test	.0000	.0000	.0879	.0000	.0000	.0000
K-S Test	.0000	.5278	1.000	.0000	.0000	.6236
South Wet						
T-test	.1031	.1931	.0002	.0746	.0002	.9853
F-test	.0000	.0000	.0000	.0000	.2856	.0000
K-S Test	.3456	.1387	.0165	.8495	.1070	.3894

Table 38. System 4 Reduction in Engagement Time Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Wet						
T-test	.0000	.0000	.0000	.0810	.4114	.9420
F-test	.0000	.0000	.4585	.1720	.0000	.0000
K-S Test	.0000	.0000	.0004	.0583	.0095	.0030

6.3.7 Reduction in Missiles Launched

Table 39. System 1 Reduction in Missile Launches Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Wet						
T-test	.4261	.0000	.0000	.0000	.0000	.0000
F-test	.1800	.0009	.0001	.0108	.0000	.0000
K-S Test	.8040	.0000	.0000	.0000	.0000	.0001
South Wet						
T-test	.0066	.6058	.0006	.0025	.0000	.0001
F-test	.0000	.0829	.0012	.0000	.0000	.0000
K-S Test	.6288	1.000	.0139	.5948	.0000	.0038

Table 40. System 3 Reduction in Missile Launches Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Wet						
T-test	*	.4302	.3034	*	*	.0464
F-test	*	.0000	.0903	*	*	.0000
K-S Test	1.000	1.000	.3843	1.000	1.000	.4677
South Wet						
T-test	*	*	*	*	*	*
F-test	*	*	*	*	*	*
K-S Test	1.000	1.000	1.000	1.000	1.000	1.000

NOTE: *Phase 2 north, HITL, Phase 2 and Phase 3 south data sets contain no standard deviation, so T-test and F-test do not provide P-values.

6.3.8 Missile Miss Distance

Table 41. System 1 Missile Miss Distance Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Dry						
T-test	.2462	.5896	.0217	.1146	.0448	.0168
F-test	.0000	.3135	.0000	.0000	.0000	.0000
K-S Test	.7176	1.000	.0000	.4789	.0000	.0000
South Dry						
T-test	.1248	.1240	.9036	.9566	.1270	.1262
F-test	.0000	.0000	.0849	.0316	.0000	.0000
K-S Test	1.000	.7230	.1596	.4860	.8392	.0324
North Wet						
T-test	.8854	.0449	.0000	.0985	.0000	.0002
F-test	.0000	.0000	.0000	.1416	.0000	.0000
K-S Test	.0000	.0021	.0000	.0856	.0000	.0082
South Wet						
T-test	.0002	.0043	.0000	.0000	.0000	.0021
F-test	.0646	.0000	.0000	.0000	.0000	.0000
K-S Test	.0000	.2064	.0000	.0000	.0000	.0000

Table 42. System 2 Missile Miss Distance Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Dry						
T-test	.2696	.0005	.0000	.0002	.0000	.9392
F-test	.0061	.0000	.0000	.0000	.0000	.0029
K-S Test	1.000	.0009	.0000	.0002	.0000	.0229
South Dry						
T-test	.7536	.0374	.0000	.0214	.0000	.1779
F-test	.0001	.0000	.0002	.0000	.0000	.1381
K-S Test	1.000	.0001	.0000	.0000	.0000	.0000
North Wet						
T-test	.0215	.7774	.1202	.0210	.0499	.1373
F-test	.1283	.0007	.0000	.0000	.0000	.0000
K-S Test	.0324	.0036	.0362	.0000	.0000	.0020
South Wet						
T-test	.0042	.8929	.1781	.0039	.0206	.1596
F-test	.0000	.1788	.0000	.0000	.0000	.0000
K-S Test	.0660	.0000	.0546	.0000	.0001	.0000

Table 43. System 3 Missile Miss Distance Correlation Matrix

Test	P3-P2	P3-HITL	P3-OAR	P2-HITL	P2-OAR	HITL-OAR
North Dry						
T-test	.5657	.0011	.1979	.0099	.1904	.1592
F-test	.3637	.0355	.0000	.0781	.0000	.0000
K-S Test	.0730	.0054	.0005	.1746	.0000	.0000
South Dry						
T-test	.6770	.1879	.0000	.0786	.0000	.0000
F-test	.4589	.0835	.0002	.0609	.0001	.0105
K-S Test	1.000	.1140	.0000	.2228	.0000	.0156
North Wet						
T-test	.3238	.0000	.0000	.4125	.9685	.0000
F-test	.0000	.0000	.0000	.0000	.0000	.0000
K-S Test	1.000	.0365	.0000	.0000	.0000	.0000
South Wet						
T-test	.2478	.0000	.0090	.6817	.1014	.0064
F-test	.0000	.0000	.0000	.0000	.0000	.0000
K-S Test	.5741	.0000	.0000	.0000	.0000	.0000

6.3.9 Conclusions

Close correlation between data sets from different test phases distinguished by P-values greater than the tester's (or reader's) level of risk suggest that the same EW Test performance measure population data can be studied successfully using the test techniques employed in either phase. Lack of correlation indicates that there is some unaccounted variable present in the test process in one or both phases that impacts that particular EW Test performance measure. Further study of other potentially impacting variables points to strong differences in data sets because of operator variance and differing threat representations between facilities.

6.4 ADS Effects on EW Test MOP Summary

Table 44 summarizes the effects of ADS on the ten different MOPs. The table covers the general effects of data latency, data loss, data corruption, and operator variance. The last column also discusses methods used to circumvent the problems encountered with an ADS test for this MOP. For a more detailed explanation of the ADS effects on each MOP, please refer to the *EW Test Classified Results Report.*

Table 44. ADS Effects On MOP Results

MOP	High Latency	Data Loss	Data Corruption (Message Changed by ADS Architecture)	Operator Variance	Tools or Methods to Minimize
Correct Threat ID	No JADS effect because threat ID is based on pod response to threat without respect to time.	If data packet is lost, ID may be missed.	If threat ID is corrupted in packet, data may be lost or false.	No impact.	Send threat ID messages reliable, and obtain text files of digibus monitor data for quality control of data.
Correct Threat ID Response Time	Delays in mode changes will affect apparent response time in real hardware.	If ID message is lost, no response time will be given.	If header time is corrupted, response time value will be false.	If not slaved to target at beginning of engagement, response times may be erroneous. (Potential exists to calculate response time based on received power that can be affected by the tracking error; potential ADS problems discussed under tracking error.)	Use special instrumentation to remove latency from data samples. Use header time to correct for latency in DSM applications. Use unmanned or pedestal- slaved threats to eliminate operator variance.
Correct ECM Technique Selection	No JADS effect because ECM technique is based on pod response to threat without respect to time.	If data packet is lost, ID may be missed.	If ECM ID is corrupted in packet, data may be missed or false.	No impact.	Send ECM ID messages reliable, and obtain text files of digibus monitor data.

MOP	High Latency	Data Loss	Data Corruption (Message Changed by ADS Architecture)	Operator Variance	Tools or Methods to Minimize
Correct ECM Technique Selection Response Time	Delay in mode changes will affect apparent response time in real hardware.	If ECM message is lost, no response time will be given.	If header time is corrupted, response time value will be false.	If not slaved to target at beginning of engagement, response times may be erroneous.	Use special instrumentation to remove latency from data samples. Use header time to correct for latency in DSM applications. Use unmanned or pedestal- slaved threats to eliminate operator variance.
Jamming-to-Signal Ratio	No JADS impact because J/S values are based on position to threat only at AFEWES. (Potential exists for errors to be introduced if AFEWES threat actions are combined with aircraft position in a different facility to determine the result. This potential exists for measured data derived from antenna patterns, transmitted powers, pointing angles, and platform positions where the positions are not in the same facility/frame of reference.)	If many samples are lost, J/S curve will look poor. (For measured J/S, insight into the quality of other measures is compromised.)	If many samples are corrupted, curve will look poor. (For measured J/S, insight into the quality of other measures is compromised.)	Since values are calculated at AFEWES regardless of tracking error, no JADS impact. For measured J/S poor tracking will produce low or undefined J/S values.	Use real-time analysis methods to watch data as they arrive and find anomalies. Ensure aircraft and threats are in same reference frame.

MOP	High Latency	Data Loss	Data Corruption (Message Changed by ADS Architecture)	Operator Variance	Tools or Methods to Minimize
RMS Tracking Error	No JADS impact by latent data because dead reckoning algorithm at AFEWES negates most latency effects. (Potential exists for errors to be introduced if threat-pointing angles are combined with aircraft position in a different facility to determine the result. Spike can be introduced into data as dead reckoning position is replaced by "actual" position.)	No JADS impact by lost data because dead reckoning algorithm at AFEWES negates most data loss effects. (Potential exists for errors to be introduced if threat-pointing angles are combined with aircraft position in a different facility to determine the result. Spike can be introduced into data as dead reckoning position is replaced by "actual" position.)	Potential exists that corruption of samples will invoke the dead reckoning algorithm or become spikes in tracking error as aircraft is moved to the incorrect position. Potential exists that data loss could prevent recovery of tracking error where threat-pointing angles are combined with aircraft position in a different facility.	Operators have significant variance in manual modes. Abnormally poor tracking can skew results.	Use as much computer control (AUTO mode) as possible. Dead reckoning algorithms assist in minimizing latency and loss effects. Use real-time analysis methods to determine anomalies in data. Ensure aircraft and threats are in same reference frame. Depending on the threat, AUTO mode may perform worse than an experienced operator.

MOP	High Latency	Data Loss	Data Corruption (Message Changed by ADS Architecture)	Operator Variance	Tools or Methods to Minimize
Number of Breaklocks (B/L)	No JADS impact because breaklocks are based on mode changes, not time of mode changes. (Potential exists for errors to be introduced if breaklocks are determined by exceeding tracking error thresholds and tracking error is derived from threat pointing angles which are combined with aircraft position in a different facility. Potential exists for breaklocks to be induced as dead reckoning position is replaced by "actual" position.)	If mode changes are lost, data samples may be inaccurate. (Potential exists for errors to be introduced if breaklocks are determined by exceeding tracking error thresholds and tracking error is derived from threat pointing angles which are combined with aircraft position in a different facility. Potential exists for breaklocks to be induced as dead reckoning position is replaced by "actual" position.)	If mode change information is corrupted, data samples may be false. Corrupted tracking error may affect this measure if breaklocks are determined by exceeding tracking error thresholds and tracking error is derived from threat pointing angles which are combined with aircraft position in a different facility.	If operator changes between AUTO and MANUAL modes inconsistently, MOP will be impacted. Abnormally poor tracking will affect this measure if breaklocks are determined by exceeding tracking error thresholds.	Use reliable transmission of mode changes, and train operators so reactions and operations are consistent. Ensure aircraft and threats are in same reference frame.
Reduction in Engagement Time	No JADS impact because engagement time is based on mode and tracking error. (Potential exists for tracking error and breaklock problems discussed above to affect the result of this measure.)	Not directly impacted by data loss, but effects on tracking error can cause engagement time to be affected.	Unless tracking error and mode changes are severely impacted, corruption will have little to no impact.	Consistent operator action is key. Variance will severely impact this MOP. Abnormally poor tracking or increases in breaklocks will skew results.	Ensure operator training to minimize fluctuations in operator reactions. Ensure aircraft and threats are in same reference frame.

MOP	High Latency	Data Loss	Data Corruption (Message Changed by ADS Architecture)	Operator Variance	Tools or Methods to Minimize
Reduction in Missiles Launched	No JADS impact because MOP is based solely on missiles fired.	Lost missile performance message will affect MOP. If TSPI drops for extended period causing B/L and tracking error to be affected, this MOP will be affected as well.	No impact unless corruption in other data causes B/L or large tracking error values.	Operator actions are key to this MOP. If firing patterns are done inconsistently, MOP will be affected. Abnormally poor tracking or increases in breaklocks will reduce valid shot opportunities.	Ensure consistent operator actions.
Missile Miss Distance	No JADS impact since missile and aircraft are in same reference at AFEWES. (Potential exists to combine missile flight path in one reference to aircraft in another reference. Also may be affected when tracking error is affected by latency as discussed above. Finally, changes in jamming onset may alter effectiveness of some techniques against some systems)	Unless missile performance message is lost, MOP can only be impacted by tracking error in JADS. (Potential exists to combine missile flight path in one reference to aircraft in another reference. Also may be affected when tracking error is affected by data loss as discussed above. Finally, changes in jamming onset may alter effectiveness of some techniques against some systems.)	Corruption of missile performance messages can cause MOP to be corrupted in JADS. Likewise, corruption of tracking error or aircraft position can affect MOP when missile flight path in one reference to aircraft in another reference are combined to produce the MOP.	Missile launch during bad track can cause large variance in data samples. Aborting missiles midflight and not reporting them also causes large variance in data samples.	Ensure consistent operator actions. Use common reference frame for aircraft and missile.

6.5 Conclusions

Overall, the ADS effects on the MOPs used in this test were far overshadowed by the human factors of the AFEWES threats. Latency can cause problems in time-critical information, but this was only a small problem in this test. Data loss, however, caused a much larger problem in the

MOPs mainly because of discontinuity in the aircraft profile. The dead reckoning algorithms used at AFEWES aided this, but they need to be modified to further minimize the problem of data loss. Lastly, data corruption did not pose a noticeable problem once test execution began.

7.0 Data Repeatability Analysis

In testing, even under the most tightly controlled conditions, it is likely that the results of a particular test, when repeated, will differ slightly. This variation is due to the statistical nature of the performance of man-in-the-loop systems, noise in electronic systems, and difficulty in replicating human and environmental conditions. These differences may be small in a well-established test facility, but nonetheless, they are not zero. For the EW Test, repeatability analysis was performed after collected SUT performance data were correlated across phases to explain why some of the MOP data sets did not match up as expected.

Some of the poor correlation experienced between phases was attributed to slight differences in threat system representation or operator practices at the various facilities. Process, equipment, and environment changes were also factors contributing to the inability to obtain correlation between data sets from different phases and facilities. Repeatability analysis provided insight into the variations that occurred between individual runs on a single day and runs performed across test days. This analysis supported the conclusion that data collected from an individual facility or during a particular EW Test phase were not representative of true system performance.

7.1 EW Test Measure of Performance Repeatability Evaluation

In performing repeatability analysis for the EW Test MOP performance data collected, no specific rule determined what was repeatable enough. Instead, the objective in analyzing each collected data set was to make an engineering assessment as to whether the sample faithfully characterized the true population of data it was collected to represent or whether there was some process change or other source of variability that could be identified.

7.1.1 Summary Statistics Review

Given this goal, repeatability analysis was performed for the EW Test SUT MOP data using a combination of evaluation techniques including review of generated summary statistics and visual assessment of plotted data across time. Each step was designed to highlight potential inconsistencies in the data. The first step identified any data points not within expected boundaries, as well as identifying sample data sets that did not follow an expected distribution (e.g., binomial, normal). This was accomplished through study of the data set minimum and maximum values, as well as the range and variance of the data values collected. These statistics are presented in the *EW Test Classified Results Report* in summary statistics tables separated by threat system, phase, and reference test condition. Analysts compared these sample statistics to expected boundaries to ensure that the sample range and distribution of values were consistent for that performance measure. Unusual variance in the data, including either extreme lack of variance or the existence of extreme outlying data points, was further researched to determine if anomalous system behavior had occurred.

7.1.2 Consistency Assessment

The next step evaluated the consistency of the data collected over time. Changes in range or variance across runs or days pointed analysts to potential system, environment, or process changes including operator learning. Visual analysis, through scatter plotting of the data collected versus time, was an indispensable technique for examining the consistency of data behavior and identifying any parameter values or trends that seemed out of line with the norm.

Figure 17 shows threat system 2 correct ECM technique response time values collected over time during Phase 3, an excellent example of how repeatable data should appear when plotted. Note that the occurrence of high and low values was consistent and within expected range boundaries across runs.

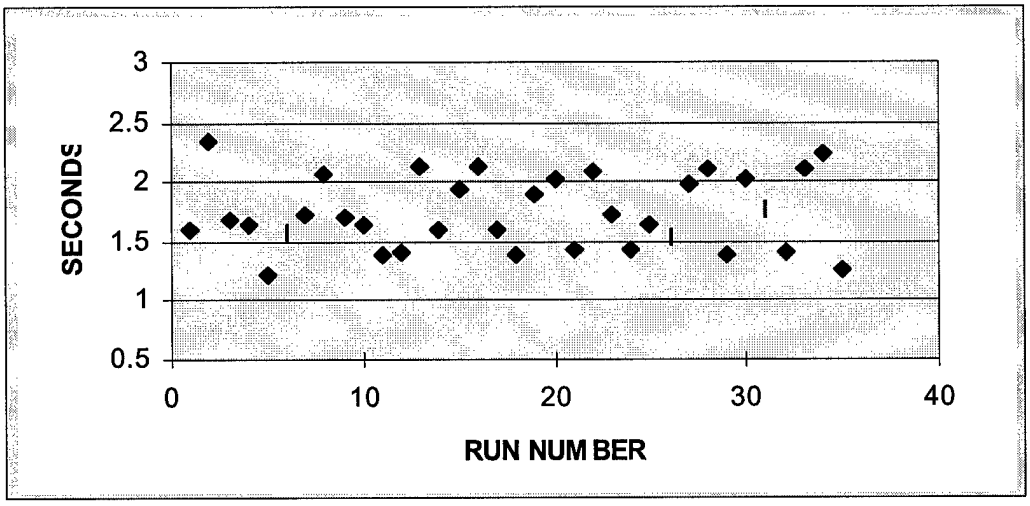


Figure 17. System 2 Correct ECM Technique Response Time - Phase 3 Southbound

Trends and anomalies, such as outliers or changes in average value or variability, were easily identified from such plots. Where inconsistencies or patterns in the data were found, analysts attempted to trace the source to some unusual test behavior by consulting facility subject matter experts and written documentation in test execution logs. Although analysts met with success in a few cases, it is important to note that not all unusual looking data could be explained.

For instance, Figure 18, shows data collected for the same threat during Phase 2 and indicates problems with repeatability. The range of collected data values was smaller than expected for this performance measure, and the outlying high and low values cast suspicion on characterized mean and variance values. Further research identified a process problem with the DSM used in Phase 2 that resulted in the generation of an atypically low variance stream of response time values. The cause of the outlying values was not determined.

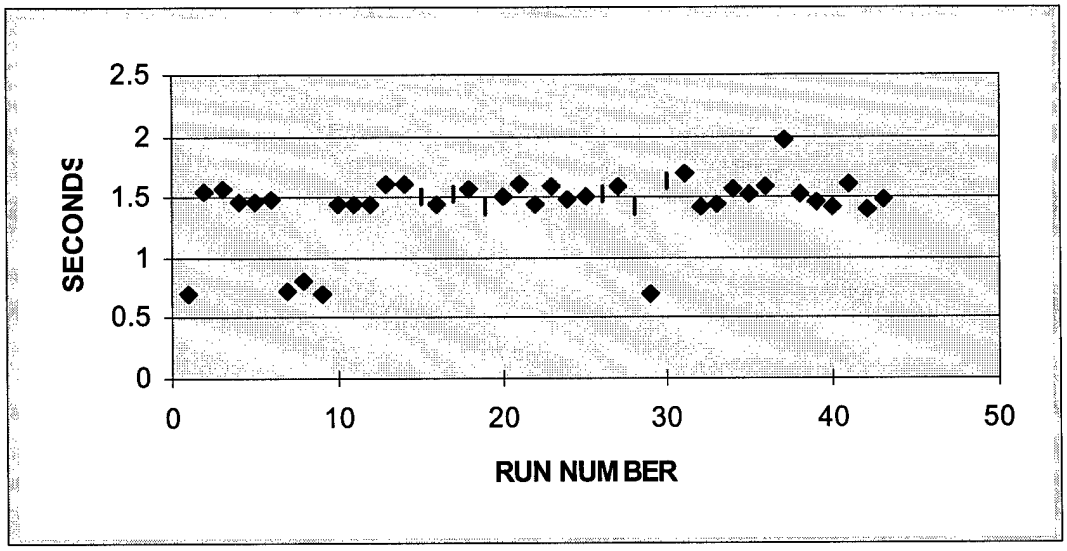


Figure 18. System 2 Correct ECM Technique Response Time - Phase 2 Southbound

Figure 19 shows correct ECM technique response time data collected over time for threat system 1 at the system integration lab during Phase 1. The lower range and average value of the last fifteen data points collected seemed to highlight some change in test process or conditions. Further research identified a system change to repair an overheated component. Real-time assessment during test execution did not identify the trend in the data samples being collected.

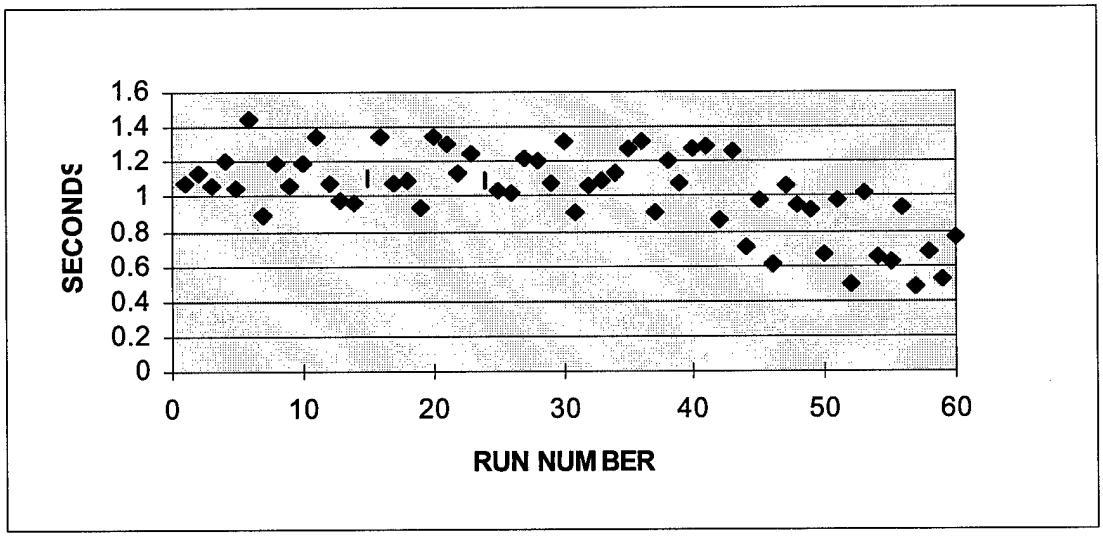


Figure 19. System 1 Correct ECM Technique Response Time - Phase 1 Southbound

7.1.3 True Population Characterization Assessment

As mentioned, there was no irrefutable guideline to follow in determining if sample data were good enough; the data simply must be repeatable enough to allow confident statements to be

made about the true population from which they came. Since EW Test correlation analysis was performed using statistical hypothesis testing on distribution shape, location (mean) and dispersion (variance) parameters, any inconsistencies in the data skewed these particular parameters and usually negatively impacted correlation.

Thus, as a final step, EW Test analysts attempted to judge how well each true performance population had been characterized by the sample data collected. Two additional statistical parameters, the standard error of the mean and the sample size, were utilized. These statistics are also presented in the summary statistics tables in the *EW Test Classified Results Report.*

The standard error of the mean was a statistic that characterized the goodness of the sample mean calculated for a data set as an indicator of the true population mean. Based on the number of samples and variability, it was a measure of the distance on either side of the sample mean within which the true population mean should fall with some particular likelihood. Sample size and variance were important aspects of this calculation, as the impacts of inconsistent (highly variable) data behavior had more influence on a small data set than a large one. Alternatively, more samples were required to gain confidence in parameters estimated from a high variance data sample than a low one. Standard error values can be computed to show the goodness of many calculated sample statistics depending on the type and distribution of the data. The standard error of the mean was calculated as follows.

$$\text{standard error}_{\text{mean}} = \frac{s}{\sqrt{n}}$$

where: s = sample standard deviation (square root of variance); and
n = number of samples

If the standard error was fairly small, then the true population mean likely fell very close to the calculated sample mean, and the true population of performance data has been fairly well characterized. Naturally, if the standard error was fairly large, the reverse was true, and conclusions drawn about the population from characterized parameters may be inaccurate. Engineering assessment was required to determine what level of characterization was necessary, depending on the particular performance measure and how the data would be utilized. Anomalous data points, if judged to have occurred outside specified test conditions, were excluded from the usable data set.

7.2 Repeatability Results

Tables 45 through 48 summarize the repeatability of the MOP data collected for each threat system across phases.

For the following tables, use this legend to identify each MOP:

ID = Correct Threat ID
ID RT = Correct Threat ID Response Time
ECM = Correct ECM Technique Selection
ECM RT = Correct ECM Technique Selection Response Time
RMS TE = RMS Tracking Error
J/S = Jamming-to-Signal Ratio
#B/L = Number of Breaklocks
RED E/T = Reduction in Engagement Time
RED # Shots = Reduction in Number of Missiles Launched
MMD = Missile Miss Distance

Table 45. Threat System 1 Data Repeatability

	OAR	HITL	PHASE 2	PHASE 3
ID	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.
ID RT	Repeatable.	N/A	Extremely tight range and lack of variance because of inappropriate use of random numbers in DSM.	Repeatable, but data value range slightly higher than OAR.
ECM	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.
ECM RT	Values for last fifteen runs significantly lower than others.	N/A	Extremely tight range because of inappropriate use of random numbers in DSM with the exception of several extreme outlying data points.	Repeatable, some data values slightly higher than OAR.
RMS T/E	Wet run data value range for OAR and HITL was significantly higher and wider than ADS phases; wet and dry data were severely impacted by extreme outlying values.		Wet run data value range for ADS phases was significantly lower and tighter than for OAR and HITL; wet and dry data were severely impacted by extreme outlying values.	
J/S	Repeatability not evaluated because of format of captured data			
# B/L	Repeatable integer data with more variance than other phases.	Repeatable integer data with no variance for dry data values.	Repeatable integer data, tightest, lowest range of wet data values.	
RED E/T	Fairly consistent but slightly wider range than other phases; primarily positive with occasional negative value.	Repeatable.	Range encompasses slightly negative to moderately positive values; sign duality impacts standard error.	
RED # SHOTS	Fairly consistent but more variance than other phases; primarily positive with occasional negative value.	Predominantly positive values with scattered negative values; sign duality impacts standard error.	Predominantly negative reduction values with scattered positive values; sign duality impacts standard error.	
MMD	Wet and dry data plagued by extreme outlying data values and high variance.	Outlying data values tend to group across several runs.	Wet and dry data fairly repeatable except for an occasional extreme outlying data value.	

Table 46. Threat System 2 Data Repeatability

MOP	OAR	HITL	PHASE 2	PHASE 3
ID	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.
ID RT	Repeatable.	N/A	Extremely tight range because of inappropriate use of random numbers in DSM with the exception of several extreme outlying data points.	Repeatable, but data value range slightly higher than OAR.
ECM	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.
ECM RT	Values for last fifteen runs significantly lower than others.	N/A	Extremely tight range because of inappropriate use of random numbers in DSM with exception of several extremely low outliers.	Repeatable, some data values slightly higher than OAR.
RMS T/E	Wet and dry data repeatable.			
J/S	Repeatability not evaluated because of format of captured data.			
# B/L	N/A	N/A	N/A	N/A
RED E/T	N/A	N/A	N/A	N/A
RED # SHOTS	N/A	N/A	N/A	N/A
MMD	Wet and dry data fairly repeatable except for a few extreme outlying data values that raise variance.	Wet and dry data fairly consistent with more "0" values than other phases.	Wet and dry data fairly repeatable except for occasional extreme outlying data values.	

Table 47. Threat System 3 Data Repeatability

MOP	OAR	HITL	PHASE 2	PHASE 3
ID	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.
ID RT	High variance compared to other phases, scattered extreme values.	N/A	Extremely tight range and lack of variance because of inappropriate use of random numbers in DSM.	Repeatable with a few outlying data values.
ECM	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.
ECM RT	Repeatable.	N/A	Extremely tight range because of inappropriate use of random numbers in DSM.	Repeatable with a few outlying data values.
RMS T/E	Wet and dry data values were trendy and inconsistent across runs. Data sets were marked by both moderate and extreme outlying values, some due to noted atypical operator/threat system mode usage.			
J/S	Repeatability not evaluated because of format of captured data.			
# B/L	Wet and dry integer data repeatable.	Wet and dry integer data repeatable.	Wet and dry integer data repeatable.	Wet and dry integer data repeatable.
RED E/T	Data ranges encompassed both negative and positive values with some extreme outlying values in both directions; sign duality impacts standard error.			
RED # SHOTS	Repeatable with little variance.	Repeatable with little variance.	Repeatable with no data variance.	Repeatable with little variance; a few atypical positive values.
MMD	Repeatable, northbound range wider than other phases for wet data.	Wet and dry data repeatable.	Wet and dry data fairly repeatable except for occasional extreme outlying data values.	

Table 48. Threat System 4 Data Repeatability

MOP	OAR	HITL	PHASE 2	PHASE 3
ID	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.
ID RT	Repeatable.	N/A	Extremely tight range and lack of variance because of inappropriate use of random numbers in DSM.	Repeatable with a few outlying data values.
ECM	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.	Repeatable with no data variance.
ECM RT	N/A	N/A	N/A	N/A
RMS T/E	Wet run data impacted early by outlying values.	Repeatable.	Wet data trendy and inconsistent; showing fewer extreme data points for later runs.	Wet run data impacted early by outlying values, some because of noted atypical operator/threat system mode usage.
J/S	Repeatability not evaluated because of format of captured data.			
# B/L	Wet and dry integer data repeatable with slightly wider range of values than other phases	Wet and dry integer data repeatable.	Wet and dry integer data repeatable.	Wet and dry integer data repeatable.
RED E/T	Repeatable, but data ranges encompass both negative and positive values.	Repeatable. Predominantly positive data values.	Repeatable. Predominantly positive data values.	Repeatable. Predominantly positive with a few outlying negative values.
RED #SHOTS	N/A	N/A	N/A	N/A
MMD	N/A	N/A	N/A	N/A

7.3 Conclusions

As described in the correlation assessment, human interactions from the AFEWES threats provided the largest problem for collecting repeatable SUT data. For the MOPs with little to no human interactions, repeatability was generally good. The MOPs that were largely dependent on human interactions provided the least repeatability in the SUT data. Even though the ROEs were designed to minimize the impact of human interactions, this rigidity did not always succeed. Future testers should seriously consider weighing the value of collecting repeatable data versus data from realistic human interactions.

8.0 Lessons Learned

JADS completed two ADS-based tests (Phase 2 and Phase 3) using an EW SUT and EW T&E systems and methods to collect MOP data. The lessons learned by JADS during pretest, test execution, and post-test phases for both tests are consolidated in this section.

8.1 Lesson 1 - Software Acceptance Testing Was Inadequate

Problem statement: Software acceptance testing of ADS components in their stand-alone mode did not uncover problems encountered once they were integrated into the ADS environment.

Impact to JADS tests: Software acceptance testing was not planned as part of the software development efforts. Formal testing was thought to be too costly and too late in the development process to be effective. JADS planned to use in-process reviews with each developer to gain insight and cross communication to get the right software products developed. However, when JADS was unable to gain insight into the software development and received obvious indications that there were flaws in some of the software items, JADS elected to use acceptance testing. Because of cost and schedule constraints, the scope of these tests was limited to the development environments and to the test sets that were available at the time. These acceptance tests did not address all software requirements. For example, the acceptance test did not consider the operational modes of the jammer digital system model as executed in the ADS environment. The acceptance testing also did not stress the model to the level of execution encountered within the JADS ADS test environment. This resulted in a model that functioned well in stand-alone mode but was marginal when integrated into the ADS environment and operated according to the test procedures.

Acceptance testing provided a more solid basis for V&V efforts. The limited acceptance test addressed several key requirements such as correct calculation of received power and correct calibration. The results of the acceptance test were available to the accreditation board to determine if the software met JADS' needs.

Finally, acceptance testing allowed a convenient point to establish configuration baselines and to transfer control of those baselines to JADS.

Corrective action taken by JADS: Acceptance testing was better planned in Phase 3 even though we still had limited test cases and tools. The new software was acceptance tested as part of the V&V plan. Formal baselines were established after completion of the acceptance tests.

Implication to future ADS-based tests: Acceptance testing of federate software is a recommended practice. These acceptance tests should be designed to (1) test the software in its intended mode of operation, and (2) test all requirements of the software. Acceptance tests can encourage the developer to fix problems before they impact the test; they also provide an excellent mechanism for supporting the V&V of the federation by proving the federates are built

correctly and satisfy the needed simulation requirements, and acceptance tests provide a clear event to which configuration management milestones can be tied.

8.2 Lesson 2 - Abbreviated Statements of Work and Distributed Simulation Testing Caused Communication Problems Between the Contractor and the Government

Problem statement: Abbreviated statement of works (SOW) and reduced deliverables resulted in differences in expectations between contractors and the government.

Impact to JADS tests: The loosely defined SOW allowed the analysis team to continue to refine requirements for a critical piece of software necessary for the test execution well beyond the date when it should have been finalized. Several measures of performance were modified from the nontraditional calculation to help measure ADS effects. This proved more difficult than expected. Since delivery schedules were not clearly defined, the contractor permitted these discussions to go on well beyond the time needed to code and test the software to meet the government expected delivery date. All parties were trying to get the best insight into ADS effects on the EW Test measures while balancing impacts to the software. The problem was resolved when the government program manager froze software requirements and provided the contractor a specific delivery date. A second impact was related to the level of on-site test support. The loosely defined SOW allowed the contractor to reallocate on-site resources earlier in the test design to support other test activities. The reallocation was discussed with the government, however, the impact to on-site support during Phase 2 was not explicitly negotiated. As a result, the government received less support than expected.

Corrective action taken by JADS: Once the software requirements were frozen, the updated software was delivered in time for test execution.

Implication to future ADS-based tests: Frequently, the government only knows what is needed in general terms to execute tests in a geographically distributed environment. Test design must mature to identify the specific capabilities that each facility will provide before specifications are created. This generally precludes creating good performance specifications prior to contract award. Sufficient tasking must be included in the SOW to ensure that government interests are covered and the lead from the contractor has a leverage tool to use on management to ensure the work is executed on time with good quality. Sequential contract awards may be used to mitigate risks associated with loosely defined SOWs.

8.3 Lesson 3 - Maintaining Schedule for an Advanced Distributed Test Execution Can Eliminate Availability Problems

Problem statement: EW tests require several critical assets to execute successfully. Delays in one of these critical assets impact the overall test schedule. This is a larger problem with ADS since delays require rescheduling multiple facilities each with unique time and asset constraints.

Impact to JADS tests: The Phase 2 test schedule slipped because of delays in obtaining data from the Phase 1 test. The DSM required response time data from the SIL test for calibration.

The first two attempts to collect these data at the OAR and HITL tests failed causing the need to perform the SIL test. Because of these previous failures, the response time data were collected much later than required to prepare for Phase 2 test execution. As a result, this test phase was delayed to properly calibrate the DSM prior to test execution.

Corrective action taken by JADS: JADS became more aggressive in managing the schedule and working with the supporting organizations to ensure that resources were ready and in place to support the test as scheduled. JADS also stated the test was not executable if it slipped again. None of the organizations wanted to be responsible for canceling the test event, so extra efforts were made by everyone to ensure the test executed successfully.

Implication to future ADS-based tests: Schedule may be the hardest factor in ADS testing to control because it is influenced by both internal factors (e.g., the ability of the different facilities to work together to identify and solve problems quickly) as well as external factors (e.g., other tests the facility must support and how much influence those tests have). Aggressive management of all development efforts and deliverables, effective risk management, and starting the effort with enough cost, schedule, and performance trade space are all essential ingredients to successful test execution.

8.4 Lesson 4 - Software Quality Assurance Is Very Important and Requires Monitoring

Problem statement: Well-defined quality software practices are important for any software development; however, when working with multiple facilities in an ADS test, strict adherence to practices is necessary to ensure success. In addition, processes for assessing software quality (e.g., independent acceptance test) were needed ensure that each ADS component operated as expected.

Impact to JADS tests: No plan existed to ensure software quality. JADS originally relied on each developer's internal practices to produce quality software. JADS attempted to gain insight into software development at each facility but failed. (See lesson 1.) Post-development quality measures were implemented to inspect delivered software. Several problems were identified with the DSM that should have been identified earlier in the software development process. Specifically, the developers sometimes misinterpreted software requirement specifications (SRS) and the ICD. These problems could have been found by closer monitoring of the software development process, particularly in the area of requirements management.

Corrective action taken by JADS: JADS became more involved in the software development process of the remaining federates. Daily contact prevented several errors from going undetected and resolved the problems before the actual test event.

Implication to future ADS-based tests: Stricter contractual requirements may be needed for organizations where the development processes are not well understood. Critical software should be developed by companies with proven subject matter experience and sound software development practices. In addition, because ADS tests will likely involve federate development

and integration in geographically distributed sites, close monitoring of and cooperation among, the individual federates is essential.

8.5 Lesson 5 - A Strong Systems Engineering Function Is Needed in ADS-Based Test Design

Problem statement: Lack of a single independent systems engineer during the development of Phase 2 software design and integration resulted in unnecessary confusion.

Impact to JADS tests: JADS assumed the lead systems engineering role throughout the JADS EW Test. During Phase 2 execution, the responsibility of system engineering unofficially transferred to other IPT members. Quite often, IPT members were also responsible for performing development tasks and delivery of several key software elements. This made it difficult for these IPT members to remain unbiased and independent during integration. The systems engineer needs to objectively identify and aggressively resolve problems. This is best done by using an independent systems engineer.

Corrective action taken by JADS: JADS reassumed the role of systems engineer as the Phase 2 test approached and during the Phase 3 test preparation.

Implication to future ADS-based tests: ADS requires strong systems integration and systems engineering. This responsibility is difficult to manage by participants supplying items to be integrated. If the sponsor is unable to provide the expertise of a systems engineer and integrator, an independent source should be used. Subject matter experience and knowledge of computers and communications technology are essential for the systems integrator.

8.6 Lesson 6 - ICD Conformance and Interpretation Problems Can Impede Completion of ADS Exercises

Problem statement: An ICD was developed for the JADS federation to guide software developers. Two problems were identified relating to this ICD: (1) nonconformance to the ICD and (2) differences in interpretation of complex concepts.

Impact to JADS tests: Prior to the test, all participants agreed that the description of the coordinate transformation was acceptable; however, facilities developed different implementations of the software when coding was finished. The problem was finally resolved when JADS provided sample transformation pairs for testing each facility algorithm. These sample data points should have been included in the JADS ICD to avoid confusion.

Test execution and post-test impacts: In some instances software was developed that did not conform to the ICD. Because of the lack of detailed acceptance testing (see lesson 1) these nonconformance problems were not found until very late in the integration process. As a result, JADS had to decide whether to bring the software into conformance or to change the ICD in order to maintain test schedule. For example, problems with the federate message sequence numbers illustrate both the test and post-test impacts. For each instance of a simulation object,

the federates should have used, in outgoing messages, sequence numbers starting at 1 and incremented by 1 for each successive message. However, because of a combination of ambiguous ICD wording and lack of early ICD compliance testing and enforcement, the TTH and AFEWES federates transmitted message sequences that did not conform to the same sequence numbering scheme.

During Phase 2 test execution this became a problem with the DSM PC real-time error checking for incoming source mode change (SMC) messages. The DSM used the sequence number to detect missing and out-of-order messages. Since the sequence numbers were not set correctly, the error reports were misleading and ineffective.

During the post-test analysis, improper message sequence numbers for several message types made it more difficult to detect and analyze runs with data loss and latency problems for the ADS analysis process. In particular, it greatly complicated the calculation of overall latencies for the critical combination of outgoing SMC messages and the corresponding jammer technique command messages generated by the DSM.

Corrective action taken by JADS: Message sequence counters were corrected for both the TTH and AFEWES threat federates for the Phase 3 test. Wording in the ICD was changed to be less ambiguous.

Implication to future ADS-based tests: Perhaps the most important lesson learned arising from the Phase 2 test and the preparations for it was the critical importance of careful planning and preparation at the earliest stages of the program. It is better to avoid problems, since there may not be enough time and/or money to find and fix them later. This seems especially true for ADS programs. The nature of ADS brings multiple facilities together, each having their own development style and practices and each bringing a potentially different understanding of the problem. This is very similar to having multiple facilities working together to develop a single software package. Any actions that reduce ambiguity in the interface design will reduce the risk of the program. This is very important for ADS-based tests since it may be difficult to slip test schedules when multiple facilities are involved. Hence, the importance of a good ICD and enforcing the same methods of compliance from the start of software development.

8.7 Lesson 7 - RTI Best Effort IP Multicast Groups Were Not Designed as Expected

Problem statement: Details on how the RTI handled its communications were withheld from the user to get them to treat the RTI as a black box. This worked for most users; however, since JADS was testing ADS, the RTI was part of the SUT. Because of this, JADS needed to know how communications were handled. JADS was surprised to learn post-test how the RTI actually created multicast groups. Instead of separate multicast groups being established according to actual publish/subscribe topology, JADS best effort data were sent in a single multicast group to which all federates were connected. Each local instance of the RTI had to deal with all messages even if its federate did not subscribe to all messages. This should have been known early in the design so that different implementations could have been tested. The following discusses how this

worked in RTI 1.3 release 4 and 5. Also there is a discussion of the data losses that were apparent and how the multicast implementation may have contributed to this.

When the RTIEXEC started execution, it transmitted Internet Group Management Protocol (IGMP) report messages to join several IP multicast groups that, for the JADS federation, had class D internet addresses of the form 224.253.xxx.yyy. The FEDEX did the same when it began, and so did each federate as it joined the federation. These IP multicast groups provided, via the UDP, the one-to-one and one-to-many best effort RTI communications infrastructure.

The RTI within each federate used the stream map in the RID file to determine to which multicast group a particular type of best effort message was sent to reach a specific federate or group of federates. The specific multicast groups joined by a federate depended on when it joined versus the other federates. Also, as new federates joined or resigned, the RTI dynamically redirected best effort traffic within the established multicast groups.

After Phase 2, JADS discovered this behavior using network packet sniffers on the SGI O_2 hosts and eventually learned from the RTI developer that the stream map in the provided RID file caused all federates joining after the third federate to stop joining new multicast groups in addition to those already created. Instead, they joined a broadcast multicast group (224.253.1.0), and federation traffic formerly sent to specific multicast groups was redirected to that group. From this, all federates received almost all best effort messages, even if they did not subscribe to them. Subsequently the local RTI component (LRC) within the federates was forced to process and discard unwanted messages. For example, the LRC in the TTH federate, which did not subscribe to threat performance messages, received, processed, and discarded five 20 Hz message streams from the platform and AFEWES federates.

During Phase 2, many instances of best effort data losses occurred that were unusual in two ways: 1) they were one-way losses, meaning that messages between two or more federates were lost in one direction but not in the opposite direction, and 2) they were selective losses. For example, the DSM federate did not receive link health, live entity state, and threat performance messages from the platform federate at JADS but received link health messages from the other three JADS federates.

These losses cannot be explained by network problems such as a short outage on one of the T-1 lines, loss of cryptographic equipment synchronization, etc., because those problems affected all best effort traffic in both directions between two test nodes. This suggested that these selective, one-way best effort data losses were due to some problem with the RTI using IP multicast groups. These losses might also come from pruning some IP multicast addresses by the protocol independent multicast-dense mode (PIM-DM) routing protocol used by the JADS EW Test routers.

Impact to JADS tests: Because of the lack of adequate documentation for the RTI RID file, JADS unknowingly used a RID file with a stream map that was probably not appropriate for a federation with six or seven federates. As a result, almost all best effort data were sent to all federates, unnecessarily loading them.

JADS experienced many unusual, selective, one-way best effort data loss events. This was possibly because of IP multicast related bugs in RTI Version 1.3 Release 4, and/or router protocol pruning of RTI IP multicast addresses. For a few runs, these events caused unacceptable response times for some DSM jammer technique commands thus reducing the number of useable data samples collected.

Corrective action taken by JADS: The *rti.rid* file could have been modified with a new stream map to provide more multicast groups to the federation. However, the latency and data loss results with RTI 1.3 Release 5 during Phase 3 integration testing were excellent so the modification was not needed. DMSO's suggestion of using data distribution management was not used because there appeared to be a significant risk of adding unacceptable latencies to the JADS EW Test real-time, performance-oriented federation.

Implication to future ADS-based tests: Federation designers need to carefully consider the instrumentation for monitoring their federations. If necessary, it may be required to carefully monitor the internal communications of the federation to adjust the networks. Phase 2 showed that RTI loggers, passive loggers, internet ping probing, and network error printouts provided only circumstantial and limited evidence to diagnose the causes of data latency and data loss problems.

RTI developers also need to document how the RTI establishes multicast groups so that federation designers can take full advantage of the RTI capabilities. High performance federations can't treat the RTI as a black box.

8.8 Lesson 8 - Reliable Test Distributor Servicing Multiple Federates Caused Unexpected Data Delays

Problem statement: When JADS began working with the RTI, complete documentation on the correct use of all the RTI services and calls was not available. JADS learned post-test that the reliable distributor servicing the federates located in the TCAC was incorrectly implemented. The following discusses the reliable distributor and how JADS implemented it for the federates in the TCAC.

Normally, every federate included a reliable distributor (RELDISTR) based on TCP, since the RTI best effort communication mechanism provided neither guaranteed delivery to all message recipients nor message delivery in order. The RELDISTR is used to send reliable data, i.e., guaranteed, in-order delivery from one federate to one or more other federates.

During the analysis of Phase 2 data loss and data delay events, there were many instances of differential latencies for reliable messages sent from a federate on one test node to two or more federates on the other nodes. For example, a latency-sensitive jammer technique command message sent by the DSM federate at ACETEF arrived with a normal latency at AFEWES and two of the JADS federates, but delayed to the other two JADS federates by hundreds to thousands of milliseconds. When the DMSO technical support for JADS was queried about such

anomalies, they advised JADS that the Phase 2 test used three RELDISTRs running on the RFENV host at the JADS node in addition to single RELDISTRs in the federates at the AFEWES and ACETEF nodes. In an effort to minimize the amount of traffic on the WAN, the DMSO liaison recommended using a single reliable distributor for the federates in the TCAC during Phase 3. This also was desirable to eliminate some types of differential latency problems. The RFENV federate was chosen to host the reliable distributor for the TCAC.

Impact to JADS tests: The RFENV federate had to be started first, since all other federates would attempt to connect to its RELDISTR. Because of the two redundant RELDISTRs in the RTIEXEC and FEDEX on the same SGI O_2 host, redundant TCP connections were apparently created (based on post-test network packet sniffer evidence) between the RELDISTRs on RFENV and those at AFEWES and ACETEF. The extra RELDISTRs and the redundant network pathways were probably the cause of some differential latency events during Phase 2.

Corrective action taken by JADS: It was determined that the RTIEXEC had its own RELDISTR, so for Phase 3 all federates in the TCAC were configured to use the RTIEXEC RELDISTR. However, because of a problem with RTI Version 1.3 Release 5, this required that the RTIEXEC be started with one version of the RTI.RID file, which then had to be replaced by a second version before the FEDEX and the RFENV federate were started. This minor inconvenience was handled by means of a UNIX shell script.

Implication to future ADS-based tests: There were two primary implications to ADS-based tests. First, federations with multiple federates on a LAN should consider using a single RELDISTR per LAN. Second, RTI developers need to clearly document how to correctly implement nondefault configurations so that federations can take full advantage of the RTI implementation features. Further implications are discussed below.

Designers, instrumenters, and executors of real-time, performance federations with latency-sensitive messages sent via the RTI reliable communications protocol to two or more federates on other distributed test nodes need to carefully consider the potential consequences of differential latencies. Differential latencies can cause the federates to have different perceptions of the simulation environment if and when critical events happen.

If an RTI developer decided to use TCP for reliable traffic, this decision can have unavoidable, long-term negative consequences that can cause trouble for some real-time, performance-oriented HLA-based simulations. For example, during the RTI performance testing leading up to the Phase 2 test, JADS learned that TCP implementations differed significantly, not only between those of different vendors, but also among different operating system version releases from the same vendor. A significant example of this was in the availability of the TCP_NODELAY option that would allow the RELDISTR TCP to acknowledge incoming TCP segments without delay. This option was not available in SGI IRIX 6.3 operating system, which was used by JADS for Phase 2 and Phase 3, but was available in IRIX 6.5, Sun Solaris and some other operating systems. Use of this option within the RTI and by the federate developers for non-HLA federate components (e.g., the DSM PC software) probably would have reduced the latencies of reliable messages.

Also, it was not at all clear that RTI developers using TCP for reliable distributor implementations had any means to guarantee that the TCP underlying a transmitting RELDISTR sent all copies of a reliable message intended for two or more recipients. Furthermore, there was no guarantee of a minimal time delay between outbound copies over separate TCP connections. The TCP protocol was never developed with this type of performance requirement in mind. Nor was it clear if intermediate RELDISTRs could introduce additional latency because of lack of control over the details of TCP actions on independent TCP connections.

8.9 Lesson 9 - Amount of RTI Reliable Traffic Can Severely Change Federation Performance

Problem statement: Federation performance varied as the mix of reliable and best effort data changed. Through trial and error, fewer problems with latency and data loss were noted if less reliable traffic was published within the federation. However, this was a subjective opinion because no tools were available to test the performance envelope of the architecture.

Impact to JADS tests: Federation performance was poor when integration testing started. The link health messages were required to be published as best effort messages to correct this problem. This change was made late in the integration effort to further tune the architecture with the real federates.

Corrective action taken by JADS: Link health check messages were published best effort during both ADS testing phases.

Implication to future ADS-based tests: Federations should experiment with different transport modes to determine the optimum mix of transport modes. RTI developers should have tools or performance measurements to guide federation developers as they design and integrate their architectures.

8.10 Lesson 10 - Time Synchronization Is Very Important but Can Not Always be Performed as Desired

Problem statement: JADS was not able to completely solve time synchronization issues using time cards in the federate computers. In theory, the hardware cards should provide the most accurate time synchronization available. In practice, some implementations proved more robust than others and verifying time synchronization across a WAN proved to be an elusive and sometimes difficult task.

The most effective configuration of the BanComm cards was initially not implemented for time synchronization on either the UNIX or PC hosts. In addition, there were problems with BanComm hardware, BanComm software, and with one contractor's attempts to write software to use the BanComm cards.

The software executing on the SGI O_2s read time directly off the BanComm cards via the JADS-developed driver software. This provided the most accurate time synchronization solution. However, the method used to obtain time information (via overloading of an IRIX operating system call) was limited because no means were available for the federates to query the BanComm card about time source. Two time sources were available: the IRIG-B time code input signal (the desired state) and the free-running internal crystal oscillator.

However, on the PCs, the BanComm software synchronized the PC system time to the BanComm card time. This was not very accurate, and in some cases, time on the PCs was incorrect by as much as 60 ms. In addition, this software did not synchronize the system time immediately when Windows 95 or Windows 98 started or restarted, which apparently caused several aborted runs because of the time on an unsynchronized ADRS PC. There was still the problem on the PCs of determining when the BanComm lost its signal and was free running on its internal oscillator.

Finally, JADS lacked an adequate method of detecting time synchronization problems in real time during federate execution runs. Only in severe cases when bursts of platform federate messages were the time synchronization problems noticed immediately and corrected.

Impact to JADS tests: Data that were time stamped on the ADRS and DSM PCs were only judged good enough. A lot of variation existed in the time values that originated on the PCs. This did not impact the ADRS PCs because the only needed time stamp was in the start command. However, the DSM PC did exhibit some odd behavior that affected calculation of jammer response times in Phase 2.

Corrective action taken by JADS: No corrective action was taken.

Implication to future ADS-based tests: If a test is going to use hardware for time synchronization (e.g., BanComm cards), obtain time directly from the hardware. Testers may need to write device drivers to enable this capability. Testers also need to resolve how they will measure time synchronization differences. Other alternatives exist in the network time protocol (NTP) software (XNTP for UNIX hosts; NTP Time for PC hosts) that provides an easy to use method to synchronize system clocks to a time source. In other JADS tests, the system clock kept within 1 ms of the time source.

8.11 Lesson 11 - Integrated Data Reduction Products Reduce Analysis Workload

Problem statement: During the data analysis of the test data, the lack of integration among data analysis products was troublesome. For the OAR test, the conversion utilities used to create files for reduction in ADRS from the OAR data files were time consuming. Furthermore, in the analysis of HITL, Phase 2, and Phase 3 data, the gathering of summary statistics and the execution of the correlation process were also very time consuming because the entire process could not be done within a single application.

Impact to JADS tests: With the additional work needed to change the data formats among the various pieces of the ADRS software, the summary statistics and graphical representations were

completed using MS Excel®. Exporting the data to MS Excel was time consuming but resulted in the ability to modify data sets and in greater flexibility in creating graphs and sorting individual data sets. Further work was required to perform correlation using *Statistix* because MS Excel did not perform the Kolmogorov-Smirnoff test. This was also very time consuming and lengthened the time needed to complete the analysis process.

Implication to future ADS-based tests: Analysts should become very familiar with the analysis software products very early in the test process. If possible, choose products that automate and complete the most work with the least amount of intervention and modification from the analysis team. If multiple products are deemed necessary, the amount of flexibility in each application to read files from other applications is very important. Some consideration should be given to building a specific analysis process that will accomplish all pieces of the analysis process within a single application. Training of analysts on the selected applications should also be accomplished early in the analysis process.

8.12 Lesson 12 - Real-Time Analysis Aids in Troubleshooting and Increases Success Rate

Problem statement: During the various test phases, real-time analysis became more and more crucial to the successful execution of the test. The largest impact on the Phase 3 test was the need to accomplish as many successful runs as possible in the least amount of test time. Without the ability to observe and critique performance from the various federation participants, the test time would have been lengthened or the useable test data collected would have been significantly decreased. ADRS and site observers were vital to correcting operator actions and clarifying the rules of engagement. SUT observers were also critical in determining if the SUT was performing as desired as the test was executed. Network observers were also needed to observe the performance of network equipment during test execution.

Impact to JADS tests: The ability in all phases to watch threat performance, operator performance, and SUT performance became a cornerstone to successful test execution. The real-time analysis supplied vital information to the test controller who could ask questions about specific equipment or operators as soon as problems were noticed. During the Phase 3 test, this capability corrected severe operator training problems at AFEWES and SUT problems at ACETEF and allowed many runs to be saved in the final data sets.

Implication to future ADS-based tests: Test design should include as much real-time analysis as possible. This will increase the success rate during test execution and minimize time required to repeat test activities if the equipment problems are not noticed until after test completion. Future testers should also consider possible actions when problems are discovered during test execution. The decision to stop testing until the problem is fixed or continue and account for the problem in the results is a difficult decision that should be considered long before test execution begins.

8.13 Lesson 13 - The Correlation Process Needs Modifications to Successfully Achieve Correlation

Problem statement: The correlation methodology used by JADS invoked strict requirements that distributions match exactly to achieve correlation. This strict requirement is not needed in the testing and correlation of EW systems because of acceptable variances in the data collection processes. The current correlation process provided very poor results when correlating the different phases of the EW Test. The statistical tests used to assess data sets means and variances were very rigid and provided for very low P-values for most data sets. The P-values calculated can lead the test manager into a false sense of failure because data sets between test phases did not correlate.

Impact to JADS tests: Based on the requirements of the correlation process, the tests could be relaxed to provide more insightful information to the test manager. For instance, when the mean miss distances for missile shots for a threat system were 24.0 and 25.5 feet between two test phases, the correlation tests gave moderately low P-values. However, the sets should be considered equal if the blast radius of the missile is 30 feet. Engineering assessments were needed to determine how much difference between two data sets was acceptable before the data sets were not considered to be from the same population. The current process did not provide meaningful insight into the threat, SUT, or operator performance, nor did it allow the tester to assess if ADS affected the MOP results.

Implication to future ADS-based tests: Future testers wishing to perform correlation should allow for engineering assessments from subject matter experts to be able to obtain useful information from the correlation process. The ultimate question between data sets collected from two different test phases is, "How much difference is too much to tolerate?"

8.14 Lesson 14 - Repeatability and Validity Are Required to Achieve Correlation

Problem statement: The correlation process assumes the collected data from the different test phases are valid and repeatable. If the data are invalid, they are not useful to the test manager to judge SUT, threat, or operator performance. If the data were not repeatable, the correlation of such data ran the risk of obtaining both false positive or false negative correlation simply because of luck of the draw with the collected data. Post-test analysis revealed that not all the data collected were repeatable. The validation process only asserted validation by the participating agencies without explicitly checking the collected data by subject matter experts. Both of these problems aided in the poor results of correlation among the different test phases.

Impact to JADS tests: Because repeatability and validity were assumed pretest and not explicitly checked, the results of the correlation process should not be used in the assessment of system performance. This was only one of many factors leading to the discredit of the correlation process and calling for modifications to future tests where correlation is used.

Implication to future ADS-based tests: System data should be validated as early in the test process as possible. Validity was the lesser of the two problems. Repeatability should be checked

during the analysis of each phase of test data. If repeatability cannot be guaranteed and proven, more data should be collected, if possible, or the analysis reports should reflect the nonrepeatable nature of the data sets. When rules of engagement or more samples can not make the data become repeatable, the correlation process should not be used on that particular data set.

8.15 Lesson 15 - MOP Definitions Require Modification to Better Assess Specific Components in an EW Test

Problem statement: During the analysis process, it became very difficult to assess the individual variance sources in the MOP data sets. Operator variance, system performance, and ADS performance, for instance, affected missile miss distance. Which of these sources caused the largest amount of variance was difficult to find and almost impossible to assert. Other MOPs, such as correct threat ID response time and correct ECM technique response time, were largely affected by data latency, but without the data sets being collected in two different manners (both with and without latency included), it was very difficult to determine if the lack of correlation was due to data latency or system performance. Many other instances were available that showed the mixture of different sources of variance and their convoluted influence on the collected data sets.

Impact to JADS tests: It was nearly impossible to point directly to the source of variance for many MOPs. This diminished JADS' ability to determine the effects of ADS on EW testing. Because the MOPs were modified to assess multiple components of the test (e.g., ADS, threat, SUT, and operator performance) the ability to comment on the specific effects from each component were greatly diminished.

Implication to future ADS-based tests: MOP definitions should be modified in future tests to assess fewer components, or the data should be collected in a manner that allows the analysis team to better determine the individual effects of each source of variance. Without the ability to perform this function, it will be troublesome to make definitive and valid statements about the ADS effects on EW testing.

8.16 Lesson 16 - Non-ADS Effects Cause the Majority of Problems for Correlation

Problem statement: Based on the analysis performed on the test data, it was discovered that non-ADS effects caused the largest amount of variance in most of the MOPs. Operator variance was the largest source of variance by far. Because the operator's choices on when to switch modes, how well to track, when to fire missiles, etc., affected the test data, constraining this was a very difficult problem. Even among the expert operators, variance from the operator was still larger than any variance caused by ADS effects such as data latency, data loss, data corruption, etc. Furthermore, threat differences and SUT differences among the different test phases also contributed to the variance in the data sets.

Impact to JADS tests: Without constraining the non-ADS effects on the MOPs collected, it was quite difficult to determine the ADS effects on the MOP data. Since the primary objective of the EW Test was to assess the ADS impacts to EW testing, the successes of this project were mostly

qualitative results based on the understanding of the MOP definitions and the qualitative results seen through the various test phases.

Implication to future ADS-based tests: For future tests, the non-ADS effects should be understood well before test execution in order to assess each component. Modifications should be made to ROE, MOP definitions, or the engagement scenario to better control the non-ADS effects or to at least be more able to separate the effects of the different sources of variance. Without the separation and control of the ADS and non-ADS effects, correlation between data sets may still prove to be an impossible task.

8.17 Lesson 17 - Most Current MOP/MOE Definitions Can Not Be Used to Assess ADS Impacts to EW Tests

Problem statement: The MOP/MOE definitions used in current EW testing allowed for many various effects to be combined into a single measure. Missile miss distance combined the effects of SUT, threat, operator, and ADS performance, which made it nearly impossible to determine the individual effect of ADS impacts. Other MOPs, such as correct threat ID response time and correct ECM technique selection response time allowed the ADS effects to be quantified, but only if the data were collected so the ADS effects could be explicitly removed from each data sample. Most other MOPs did not allow for an accurate assessment of ADS impacts to EW testing.

Impact to JADS tests: Without the ability to separate the individual effects of ADS impacts, the successful completion of the JADS tasking to determine the utility of ADS for various types of testing was weakened. Because the MOP/MOE definitions combined the different sources of variance, it was only possible to make educated guesses about the impacts of ADS to EW testing. It was possible in a very few cases to effectively determine the ADS effects on the performance of the various components, and without quantitative data to back JADS' claims, the results and interpretations of ADS utility were subject to individual opinion.

Implication to future ADS-based tests: Future testers should attempt preliminary testing to determine if the MOP definitions selected will allow the expected results to be collected from the test execution. More so, the determination of which components will be assessed should be determined very early, and the test design should be modified to accommodate these assessments.

9.0 Conclusions/Recommendations

This report described the ADS implementation, development and integration process; ADS and correlation results; and lessons learned for the Phase 3 test. Conclusions to the JADS issues are addressed. While correlation results were included, the underlying EW Test data are not presented. These results are classified and contained in the *EW Test Classified Results Report.*

The Phase 3 test used an HLA-compliant ADS architecture to successfully recreate both an OAR test and a HITL test. The Phase 3 architecture successfully integrated an ISTF representing an advanced development stage of a self-protection jammer with the high fidelity threats at AFEWES. This implied that ADS may be used to address the EW test process issues of correlation and fidelity. This report does not fully discuss the utility of ADS for EW testing. Complete discussion on the utility of ADS to EW testing is contained in *The Utility of Advanced Distributed Simulation for Electronic Warfare Testing.*

Examination of the MOP data indicated that there were isolated incidents of ADS impacting the test results. Aircraft position data dropouts seen during integration testing forced JADS to add a simple dead reckoning algorithm into the AFEWES gateway. This fix greatly helped the data dropout problem. However, when the data resumed, the aircraft was immediately moved to the updated location causing jumps in the aircraft position. This effect influenced the MOP data in various ways ranging from the bad flyout of a missile to increased tracking error and extra breaklocks. The interviews indicated that there were some odd aircraft behaviors at the start of several of the scripts (but outside the core engagement area) that operators deemed unrealistic. These did not affect the EW MOPS. The data and interviews indicated that there were no consistent ADS-induced biases or flaws that would make the data invalid.

These results met expectations given that the test design took advantage of the unique capabilities of the AFEWES facility. Properly designed ADS architectures should not impact test results. Combining events in separate facilities into a single MOP, such as correct threat ID, impacted measures where the transmission latency was significant compared to the duration of the actual event. During the Phase 2 test, it was possible to assess the effects of latency on the timing MOPs, but only because of problems in the DSM software. This task was not possible in the Phase 3 test because the variance in latency and the variance in response times could not be separated. Overall, data latency in excess of the design goal and lengthy bursts of lost aircraft position data did not affect the EW MOPS in a consistent, measurable fashion. This performance and the lack of impact were somewhat surprising. The dead reckoning algorithms were very useful and offered room for improvement in future tests to eliminate the jumps in aircraft position.

There were limitations within the JADS-created ADS architecture. Different jammer techniques and more reactive players (fewer ROE for the threats and maneuvering aircraft) required that the bursts of lost aircraft position data be resolved and latency performance be improved over what was observed in the Phase 2 test. Predictive jammer techniques would also require more of the jammer processing logic to be collocated with the JETS at AFEWES. Several of the message structures and common data used in the test require examination before being used in other tests.

While all the message structures had room for growth, they should be examined by future implementers to ensure the size and intent meet the requirement of the new federation.

The most significant limitation to this architecture was the availability of high fidelity threats suitable for ADS-based testing. Low fidelity threats were not difficult to add to this architecture. However, to address the correlation and fidelity shortfalls of the EW test process, high fidelity threat representations were the keys to the highest benefit from this architecture. The AFEWES facility used distributed simulation techniques within its facility to accomplish traditional testing. ADS simply allowed AFEWES to connect to other facilities or locations. The OAR used in Phase 1 had high fidelity threat simulators as well. However, the OAR threats were not suitable to accomplish testing within the JADS architecture. R/F injection into the threat for both target and jamming was a key to this architecture. The second key was the infrastructure to tie the threats together to engage a common virtual target in a common synthetic environment. Neither of these was available at the OAR used in Phase 1. According to the CROSSBOW-sponsored Threat Simulator Linking Activity (TSLA) study, these types of high fidelity threat simulators are very scarce resources.

Appendix A - Site Controller Matrix

Condition	Range (nmi from IP)	SADS III	SADS VI	SADS VIII	WEST X
Northbound					
1	0.0	ON	OFF	OFF	OFF
2	4.5	"	ON	"	"
3	6.0	"	"	ON	"
4	8.6	"	"	"	ON
5	13.6	"	"	OFF	"
6	16.0	OFF	"	"	"
7	17.6	"	OFF	"	OFF
Southbound					
1	1.5	OFF	OFF	ON	OFF
2	3.7	"	ON	"	"
3	6.0	ON	"	"	"
4	16.0	"	"	OFF	"
5	17.5	"	OFF	"	"
6	21.0	OFF	"	"	"

IP = initial point
SADS = Simulated Air Defense System
nmi = nautical miles
WEST = Weapon Evaluation Simulated Threat

Appendix B - Phase 3 Script Execution Matrix

When executing the Phase 3 test, Section B1 was used when the Simulated Air Defense System (SADS) VIII and Weapon Evaluation Simulated Threat (WEST) X were manned at AFEWES. The profiles listed in Section B2 list the excursion runs executed to test the self-protection jammer (SPJ) under conditions different from the reference test condition. The profiles listed in Section B3 were used to test all four live threats. Section B4 was used when executing profiles with the SADS III and SADS VI manned at Air Force Electronic Warfare Evaluation Simulator (AFEWES).

The mission and run designations were derived from the open air range (OAR) mission profiles. The profiles are numbered *XY-ZZ*.

X designates which threats were manned at AFEWES (0 or 1 = all live threats, 2 or 3 = SADS III and SADS VI live, 4 or 5 = SADS VIII and WEST X live).

Y designates the OAR mission used to generate the profile (5-9 = OAR missions 5-9, 0-1 = OAR mission 10-11).

ZZ designation is the number of the specific run from the OAR mission (1-20 = OAR runs 1-20). For example, profile 45-5 means a live SADS VIII and WEST X from OAR mission 5, run 5.

The only profiles deviating from this scheme were most of the 14 excursion runs, which were arbitrarily named.

Section G1 - SADS VIII and WEST X Live at AFEWES

The profiles in this section had the activation messages in the terminal threat hand-off (TTH) script for the SADS VIII and WEST X. The radio frequency environment (RFENV) script contained the mode messages to load the SPJ for the SADS III and SADS VI threat systems, which were not manned.

NORTH				SOUTH			
Dry	**Wet**			**Dry**	**Wet**		
45-6	45-10	45-12	46-2	45-3	45-5	45-7	45-9
46-4	46-6	46-8	46-10	46-7	46-9	46-11	46-17
46-18	47-4	47-6	47-8	47-1	47-3	47-5	47-7
47-12	47-14	47-16	47-18	47-9	47-13	47-17	47-19
47-20	48-2	48-4	48-6	47-21	48-3	48-5	48-7
48-8	49-5	50-7	50-9	49-4	50-4	50-6	50-12
50-13	50-15	51-2	51-4	50-14	50-18	51-1	51-3
51-6	51-8	51-10		51-5	51-7	51-9	

Live SADS VIII and WEST X

NORTH				SOUTH			
Dry	**Wet**			**Dry**	**Wet**		
46-2	45-10	45-6	45-12	45-7	45-5	45-3	45-9
46-10	46-6	46-4	46-8	46-11	46-9	46-7	46-17
47-8	47-4	46-18	47-6	47-5	47-3	47-1	47-7
47-18	47-14	47-12	47-16	47-17	47-13	47-9	47-19
48-6	48-2	47-20	48-4	48-5	48-3	47-21	48-7
50-9	49-5	48-8	50-7	40-6	50-4	49-4	50-12
51-4	50-15	50-13	51-2	51-1	50-18	50-14	51-3
	51-8	51-6	51-10	51-9	51-7	51-5	

Live SADS VIII and WEST X

Section B2 - 14 Excursion Runs

The following profiles were executed to test the ability of advanced distributed simulation (ADS) to handle a more erratic scenario. The description of each profile follows the mission and profile number.

	DESCRIPTION			
Mission - Profile	**Speed (knots)**	**Altitude (feet msl)**	**N-S**	**Notes**
11-1	360	9 K	N	standard rules of engagement (ROE)
11-1	360	9 K	N	standard ROE
81-1	360	9 K	N	sites come up in accordance with (IAW) site controller matrix (SCM) - simultaneous missiles at overlap
81-1	360	9 K	N	sites come up IAW SCM - simultaneous missiles at overlap
82-1	550	9 K	N	standard ROE
82-1	550	9 K	N	simultaneous missiles
83-1	720	9 K	N	simultaneous site call-up - fire at will
83-1	720	9 K	N	simultaneous site call-up - simultaneous missiles
9-5	360	9 K	S	standard ROE - aircraft (A/C) ascent to 15,000 feet
9-5	360	9 K	S	standard ROE - A/C ascent to 15,000 feet
84-1	360	6.5 K	N	standard ROE
84-1	360	6.5 K	N	standard ROE
85-1	360	20 K	N	standard ROE
85-1	360	20 K	N	standard ROE

K = thousand
N = northbound
msl = mean sea level
S = southbound

Excursion Run Matrix

Section B3 - All Live Threats at AFEWES

Profiles in this section were executed when all threats were manned at AFEWES.

NORTH				SOUTH			
Dry	**Wet**			**Dry**	**Wet**		
5-6	5-10	5-12	6-2	5-3	5-5	5-7	5-9
6-4	6-6	6-8	6-10	6-7	6-9	6-11	6-17
6-18	7-4	7-6	7-8	7-1	7-3	7-5	7-7
7-12	7-14	7-16	7-18	7-9	7-13	7-17	7-19
7-20	8-2	8-4	8-6	7-21	8-3	8-5	8-7
8-8	9-5	10-7	10-9	9-4	10-4	10-6	10-12
10-13	10-15	11-2	11-4	10-14	10-18	11-1	11-3
11-6	11-8	11-10		11-5	11-7	11-9	

All Threats Manned at AFEWES

Section B4 - SADS III and SADS VI Live at AFEWES

This section lists the profiles used when the SADS III and SADS VI were live and manned at AFEWES. The TTH scripts contained the activations and deactivations for the SADS III, and the RFENV scripts contained the modes for the SADS VIII and WEST X to load the jammer federate.

NORTH				SOUTH			
Dry	Wet			Dry	Wet		
25-6	25-10	25-12	26-2	25-3	25-5	25-7	25-9
26-4	26-6	26-8	26-10	26-7	26-9	26-11	26-17
26-18	27-4	27-6	27-8	27-1	27-3	27-5	27-7
27-12	27-14	27-16	27-18	27-9	27-13	27-17	27-19
27-20	28-2	28-4	28-6	27-21	28-3	28-5	28-7
28-8	29-5	30-7	30-9	29-4	30-4	30-6	30-12
30-13	30-15	31-2	31-4	30-14	30-18	31-1	31-3
31-6	31-8	31-10		31-5	31-7	31-9	

SADS III and SADS VI Threats Manned at AFEWES

NORTH				SOUTH			
Dry	Wet			Dry	Wet		
25-12	25-10	25-6	26-2	25-7	25-5	25-3	25-9
26-8	26-6	26-4	26-10	26-11	26-9	26-7	26-17
27-6	27-4	26-18	27-8	27-5	27-3	27-1	27-7
27-16	27-14	27-12	27-18	27-17	27-13	27-9	27-19
28-4	28-2	27-20	28-6	28-5	28-3	27-21	28-7
30-7	29-5	28-8	30-9	30-6	30-4	29-4	30-12
31-2	30-15	30-13	31-4	31-1	30-18	30-14	31-3
31-10	31-8	31-6		31-9	31-7	31-5	

SADS III and SADS VI Threats Manned at AFEWES

Appendix C - Acronyms and Definitions

412 TW	412th Test Wing, Edwards Air Force Base, Florida
A/C	aircraft
AAA	anti-aircraft artillery
AATC	Air National Guard Air Force Reserve Test Center, Tucson, Arizona
ACETEF	Air Combat Environment Test and Evaluation Facility, Patuxent River, Maryland; Navy facility
ADRS	Automated Data Reduction Software
ADS	advanced distributed simulation
AFEWES	Air Force Electronic Warfare Evaluation Simulator, Fort Worth, Texas; Air Force managed with Lockheed Martin Corporation
ALQ-131	a mature self-protection jammer system; an electronic countermeasures system with reprogrammable processor developed by Georgia Tech Research Institute
AMES	Automatic Multiple Environment Simulator, Eglin Air Force Base, Florida
AMI	alternate mark inversion
AOA	angle of arrival
APA	analysis plan for assessment
API	application program interface
ASCII	American Standard Code for Information Interchange
ATEWES	Advanced Tactical Electronic Warfare Environment Simulator
AWC	Air Warfare Center at Nellis AFB, Nevada, and Eglin AFB, Florida
B/L	breaklock
B8ZS	binary eighth zero substitution
BERT	bit error rate test
C4ISR	command, control, communications, computers, intelligence, surveillance and reconnaissance
CAIV	cost as an independent variable
COTS	commercial-off-the-shelf
CRM	communications resource manager
CROSSBOW	Office of the Secretary of Defense committee under the Director, Test, Systems Engineering and Evaluation
CSU	channel service unit
CW	continuous wave
dB	decibel
dBm	decibel millivolts
DD, DT&E	Deputy Director, Developmental Test and Evaluation
DEC	Digital Equipment Corporation
DIS	distributed interactive simulation
DISA	Defense Information Systems Agency
DISN	Defense Institute Systems Network
DMAP	data management and analysis plan

DMSO	Defense Modeling and Simulation Organization, Alexandria, Virginia
DoD	Department of Defense
dry run	the system under test is off
DSM	digital system model
DSU	data service unit
DT&E	developmental test and evaluation
DTM	digibus traffic monitor
DTMS	Digibus Traffic Monitor System
E&M	analog voice signaling standard
EAV	early access version
ECCM	electronic counter-countermeasures
ECM	electronic countermeasures
EIOB	enhanced input/output buffer
EMC	electromagnetic compatibility
EMI	electromagnetic interference
ESF	extended super frame
EW/EW Test	electronic warfare; JADS Electronic Warfare Test
EWIR	electronic warfare integrated reprogramming database
FAT	federate acceptance test
FCR	fire control radar
FEDEX	federation executive
FEPW	federation execution planners workbook
FIT	federate integration test
FOM	federation object model
FY	fiscal year
GHz	gigahertz
GPS	global positioning system
GTRI	Georgia Tech Research Institute, Atlanta, Georgia
H/W	hardware
HITL	hardware-in-the-loop (electronic warfare references)
HLA	high level architecture
HSNPL	hardware, software, and network problem log
HUD	heads-up display
Hz	hertz
I/C	interface and control
I/F	interface
I/O	input/output
IADS	Integrated Air Defense System
IAW	in accordance with
ICD	interface control document
ID	identification
IDNX™	Integrated Digital Network Exchange
IEEE	Institute of Electrical and Electronics Engineers
IGMP	Internet Group Management Protocol
INS	inertial navigation system

IP	internet protocol; initial point
IPT	integrated product team
IRIG	Inter-Range Instrumentation Group
IRIX	operating system for the Silicon Graphics, Inc.
ISTF	installed systems test facility
J/S	jamming-to-signal ratio
JADS	Joint Advanced Distributed Simulation, Albuquerque, New Mexico
JETS	JammEr Techniques Simulator
JT&E	joint test and evaluation
JTC	jammer technique command
JTF	Joint Test Force, Albuquerque, New Mexico
K	thousand
Kbps	kilobits per second
KG	a family of communications security equipment
kHz	kilohertz
KIV	AlliedSignal embedded KG-84 (a family of communications security equipment) communications security module
Kpps	1000 packets per second
K-S	Kolmogorov-Smirnov test
LAN	local area network
LHC	link health check
LRC	local runtime infrastructure component
Mb	megabyte
Mbps	megabits per second
MHz	megahertz
MOE	measure of effectiveness
MOP	measure of performance
Mpps	million packets per second
MS	multispectral
ms	millisecond
msl	mean sea level
MT	message tape
MTI	moving target indicator
N&E	network and engineering
nmi	nautical mile
NRZ	nonreturn to zero
ns	nanosecond
NSA	National Security Agency
NTP	network time protocol
OAR	open air range
OFP	operational flight program
OSD	Office of the Secretary of Defense
OT&E	operational test and evaluation
PC	personal computer
PCM	pulse code modulation

PIM-DM	protocol independent multicast-dense mode
PRI	pulse repetition interval
P-value	probability value
PT	preflight message tape
PTP	program test plan
PW	pulse width
PX	packet exchange
QA	quality assurance
QAVP	quad analog voice processor
R/P	receiver processor
RAD	company that manufactures the voice signal converter
RCS	radar cross-section
RELDISTR	reliable distribution
RF	radio frequency
RFENV	radio frequency environment
RID	runtime infrastructure initialization data
RMS	resource management system
ROE	rules of engagement
RTC	reference test condition
RTI	runtime infrastructure
RTIEXEC	runtime infrastructure executive
RWR	radar warning receiver
S/W	software
SAC	senior advisory council
SADS	Simulated Air Defense System
SAM	surface-to-air missile
SCM	site controller matrix
sec	second
SGI	Silicon Graphics, Inc.
SIL	system-in-the-loop; system integration laboratory
SMC	source mode change
SME	subject mater experts
SNMP	Simple Network Management Protocol
SOW	statement of work
SPAG	software-programmable antenna pattern generator
SPECTRUM®	a network analysis package developed by Cabletron Systems
SPJ	self-protection jammer
SRS	software requirements specification
STEP	simulation, test and evaluation process
SUT	system under test
SWEDAT	simulation warfare environment generator data file
SWEG	Simulated Warfare Environment Generator at AFEWES
T&E	test and evaluation
T/E	tracking error
T-1	digital carrier used to transmit a formatted digital signal at 1.544 megabits

	per second
TAB	technical advisory board
TAMS	Tactical Air Mission Simulator
TAP	test activity plan
TCAC	Test Control and Analysis Center, Albuquerque, New Mexico
TCF	test control federate
TCP	transmission control protocol
TFP	time and frequency positioning
TMC	test management center
TP	threat performance; tactical plot
TSLA	Threat Simulator Linking Activity
TSPI	time-space-position information
TTH	terminal threat hand-off federate
TTR	target tracking radar
UDP	user datagram protocol
UTC	universal time code
V&V	verification and validation
VSC	voice signal converter
VV&C	verification, validation and certification
WAN	wide area network
WEST	Weapon Evaluation Simulated Threat
wet run	the system under test is on
WTR	Western Test Range
Y2K	year 2000